(2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THE DEVELOPMENT OF A THERMAL ANALYSIS MODEL
BUILDER FOR A PRINTED CIRCUIT BOARD

by

Stephen J. Glaser

September 1991

Thesis Advisor: Allan D. Kraus

Approved for public release; distribution is unlimited.

Unclassified

security classification of this page

REPORT DOCUMENTATION PAGE

1a Report Security Classification Unclassified	1b Restrictive Markings
2a Security Classification Authority 2b Declassification Downgrading Schedule	3 Distribution Availability of Report Approved for public release; distribution is unlimited.
4 Performing Organization Report Number(s)	5 Monitoring Organization Report Number(s)
6a Name of Performing Organization Naval Postgraduate School 6b Office Symbol (*if applicable*) EC	7a Name of Monitoring Organization Naval Postgraduate School
6c Address (*city, state, and ZIP code*) Monterey, CA 93943-5000	7b Address (*city, state, and ZIP code*) Monterey, CA 93943-5000
8a Name of Funding Sponsoring Organization 8b Office Symbol (*if applicable*)	9 Procurement Instrument Identification Number
8c Address (*city, state, and ZIP code*)	10 Source of Funding Numbers Program Element No / Project No / Task No / Work Unit Accession No

11 Title (*include security classification*) THE DEVELOPMENT OF A THERMAL ANALYSIS MODEL BUILDER FOR A PRINTED CIRCUIT BOARD

12 Personal Author(s) Stephen J. Glaser

13a Type of Report	13b Time Covered	14 Date of Report (*year, month, day*)	15 Page Count
Master's Thesis	From To	September 1991	155

16 Supplementary Notation The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17 Cosati Codes			18 Subject Terms (*continue on reverse if necessary and identify by block number*)
Field	Group	Subgroup	Thermal model builder, printed circuit board, computer modeling.

19 Abstract (*continue on reverse if necessary and identify by block number*)

The Naval Postgraduate School possesses software designed to perform thermal analysis of electronic components. At the core of this package is a model builder whose purpose is to generate a thermal model for use in steady state and transient thermal analyzers. The current version of the model builder requires excessive amounts of time for data input and model verification. This thesis describes the development of a model builder specifically designed to reduce the time required to model a printed circuit board containing up to four copper layers.

20 Distribution Availability of Abstract ☒ unclassified unlimited ☐ same as report ☐ DTIC users	21 Abstract Security Classification Unclassified
22a Name of Responsible Individual Allan D. Kraus	22b Telephone (*include Area code*) (408) 646-2730 22c Office Symbol EC KS

DD FORM 1473,84 MAR

83 APR edition may be used until exhausted
All other editions are obsolete

security classification of this page

Unclassified

Approved for public release; distribution is unlimited.

The Development of a Thermal Analysis Model Builder for a Printed Circuit Board

by

Stephen J. Glaser
Lieutenant, United States Navy
M.S., Purdue University, 1984

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1991

Author: ______________________
Stephen J. Glaser

Approved by: Allan D. Kraus
Allan D. Kraus, Thesis Advisor

William Gragg
William Gragg, Second Reader

Michael A. Morgan
Michael A. Morgan, Chairman,
Department of Electrical and Computer Engineering

ABSTRACT

The Naval Postgraduate School possesses software designed to perform thermal analysis of electronic components. At the core of this package is a model builder whose purpose is to generate a thermal model for use in steady state and transient thermal analyzers. The current version of the model builder requires excessive amounts of time for data input and model verification. This thesis describes the development of a model builder specifically designed to reduce the time required to model a printed circuit board containing up to four copper layers.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

The Naval Postgraduate School possesses software designed to perform thermal analysis of electronic components. The software package contains two routines designed to generate a thermal model (in the form of an ASCII input data file) to be read by the thermal analyzer program. The first routine is a generalized model builder used in developmental stages, as well as an editor used to modify existing models. The second routine was designed to work with models that have a specific geometric configuration.

In order to accurately model electrical components, the structure must be subdivided into small, equal sized subvolumes. The centroid of each subvolume is referred to as a node, and due to the assumed isothermal nature of each subvolume, a node is considered as representative of the total subvolume.

Producing an accurate thermal model, requires the design engineer to deal with enormous amounts of equations and temperatures, each describing an indiviual node. Additionally, each node is connected to adjacent nodes by thermal conductances. As the desired accuracy increases, the number of nodes and equations also increases. Modeling the electronic component without the aid of a computerized model builder is a task requiring inordinate amounts of time.

Current versions of the thermal analyzer contain provisions for the generation of the node equations and node interrelationships; however, data input is still a manual, time-consuming process. At present there exists a need for a program that will generate a data-input file for the thermal analyzer that is both generated through a menu-driven routine, and allows the design engineer enough flexibility to model the electronic component to suit his or her needs.

This thesis describes the development of a model builder designed specifically to reduce the time required to model the copper and epoxy layers of a printed circuit board. A typical printed circuit board configuration is shown in Figure 1. The printed circuit board may contain up to four copper layers. Additional features of the model builder covered are:

1. The capability of working in SI or English units.
2. The choice of a total of sixteen aspect ratios.
3. The provision for up to 740 nodes.
4. The ability to specify the percent of copper coverage for each layer.

5. The ability to specify dimensional parameters for each individual copper layer.
6. The ability to input heat dissipation using several methods.
7. The provision for six ambient temperatures.
8. The automatic calculation of conductance values based on user input.

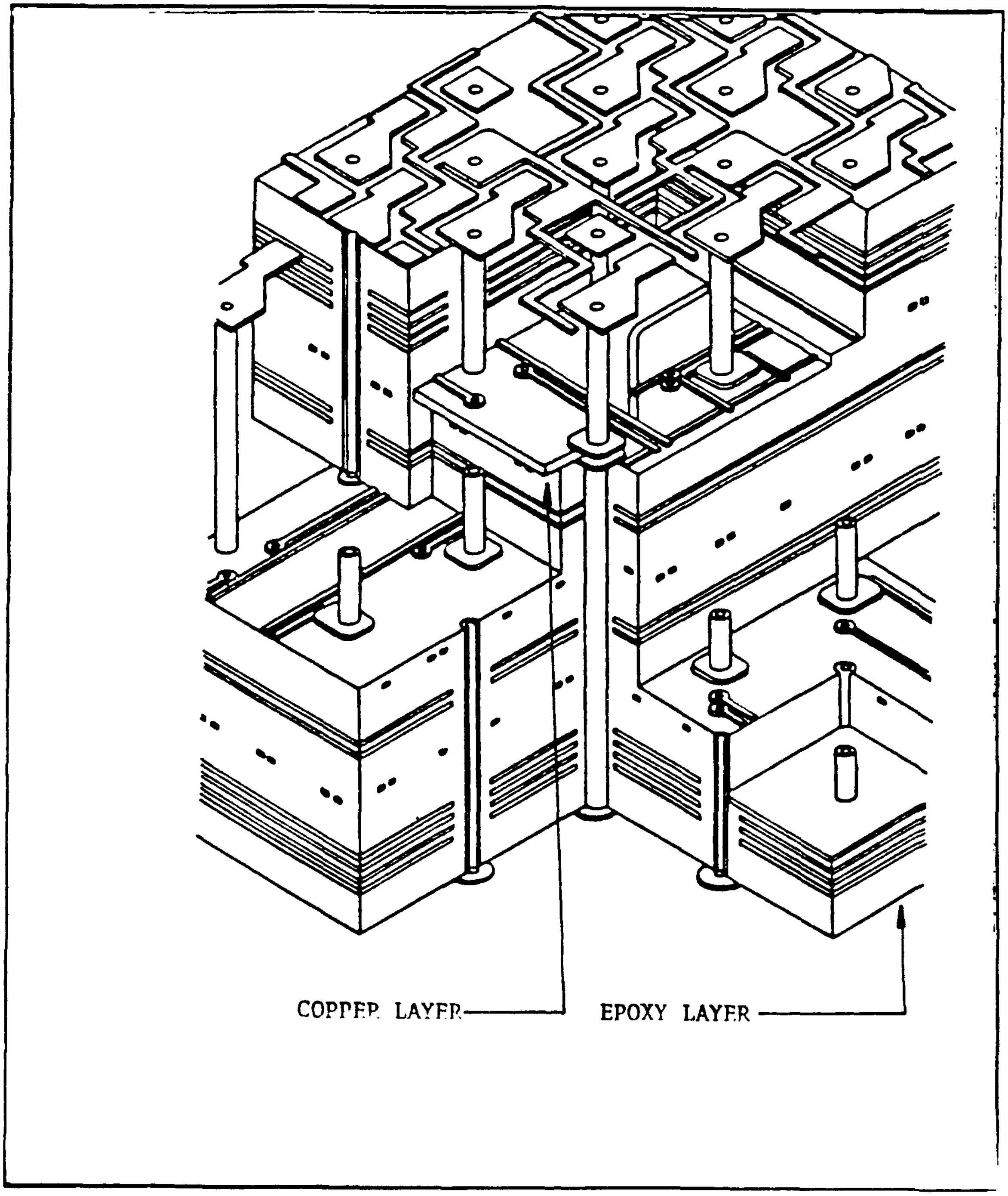

Figure 1. Typical printed circuit board configuration.
Source: Reference 2.

II. A JUSTIFICATION FOR THERMAL ANALYSIS

A printed circuit board is a conglomeration of organic and inorganic materials with external and internal wiring, which allows electronic components to be physically supported, and electrically connected. [Ref. 1]

Over the past several decades, printed circuit board technology has developed substantially. Early printed circuits were fabricated by printing a pattern of polymer resist on a copper plane and then chemically etching. Holes drilled in the laminate held the component leads that were soldered to the copper-printed patterns. The technology has progressed in developing the sophistication and uses of the printed circuit board interconnections.

Today, the basic functions of the printed circuit board are the same; the interconnecting copper signal lines join two I/O leads from two different components. The components may be resistors, inductors, capacitors, or semiconductor chips. When applying multichip technologies, there may be hundreds of components attached to the printed circuit board. The ever-increasing level of complexity of printed circuit boards has forced primitive boards using a few yards of printed wiring in the 1960s, to evolve into sophisticated, multi-layered structures requiring kilometers of printed wiring. This increase in the circuit board level of sophistication can be attibuted to the integration of semiconductors and an increased need for I/O capabilities. [Ref. 2]

The most complex printed circuit boards contain kilometers of copper interconnection, roughly 50 to 100 microns wide, and half as thick. These boards distribute KW/m^2 of power internally in very densely packed layers of copper. The drive for higher performance means that there is a greater requirement for power-handling and cooling capabilities. [Ref. 2]

It is the responsibility of the designers to ensure that cooling on the printed circuit board is adequate under all possible load conditions in order to allow proper performance of the individual components and the board as a whole. Therefore, it is imperative that the designers understand and are able to predict the temperature distribution on multilayered structures prior to prototype production. The overriding reasons for performing a precise thermal analysis are to increase component reliability, ensure proper material selection, reduce the possibility of catastrophic thermal failure, and guarantee electrical performance.

A. RELIABILITY

Reliability is defined as the probability that a component is functioning as designed, while failure is defined as the probability that a component is not functioning as designed. There is a predictable relationship between the operating temperature of electronic components and reliability. The materials used in the fabrication of these components have thermal limitations, and should these thermal limitations be exceeded, the physical and chemical properties of the material are affected, and the device fails. [Ref. 1]

For a large number of components, a typical plot of failure rate as a function of time is shown in Figure 2. Failures at short times are called early fails, or infant mortality, while failures at long times are called wearout fails because they result from usage. At all times, failures can occur from intrinsic mechanisms, or from random overstress. [Ref. 3]

Provided the device has been adequately designed, early failures can arise as a result of manufacturing defects. Defects that occur early on, or the "burn-in" period, are considered to be the result of poor or inadequate quality control mechanisms in the manufacturing process. [Ref. 2]

Of greatest concern are the failures that occur during the useful life of the device because the probability of failure during this period should be nearly zero. Should a device fail during its useful life, the probable cause of failure would be due to a variety of external factors, and are unpredictable. [Ref. 2]

As time and usage progress, the terminal period of wearout is encountered, usually well past the the end of system life (EOL), in which the failure rate increases. Conductor electromigration is a typical example of a wearout mechanism, in which the electron flow itself causes irreversible mass flow, which causes the formation of voids and consequent conductor failure. Printed circuit boards seldom have significant failure rates when they are produced with sufficient characterization and control. When failures do occur, they can normally be attributed to either manufacturing defects or lack of integrity to the design specifications. The most notable concerns are the resin and laminate effects that can give rise to insulation integrity. As printed circuit boards become more densely packaged and continue to spread into more unconventional environments such as homes, automobiles or marine engines, there is a need to establish improved materials and process controls for even better reliability. Thermal analysis of the printed circuit board becomes one of the fundamental facets of the design process in order to better characterize the printed circuit board. [Ref. 2]

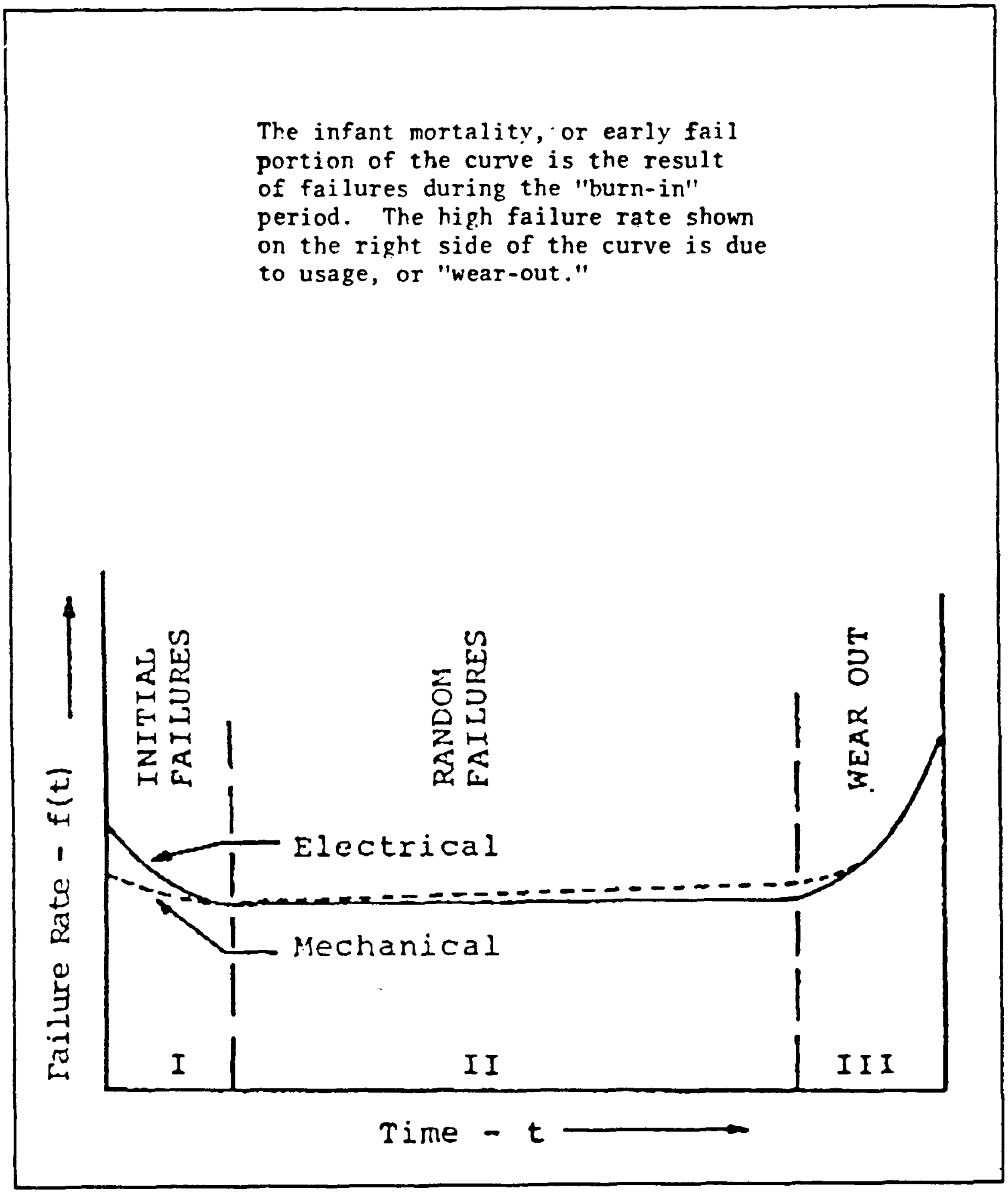

Figure 2. Failure rate as a function of time for typical component.

Source: Reference 3.

B. MATERIAL

The fabrication of printed circuit boards results in the joining of different materials. The materials selected can have a significant impact in thermal properties of the printed circuit board. In the largest circuit boards hundreds of amps may be switched at once. As packaging densities increase thermal, mechanical, electrical, and chemical coupling becomes very strong. In view of the complexity of electrical structures today, computer modeling of total thermal response of a printed circuit board is a requirement conducive to understanding one of many interactions. [Ref. 2]

C. THERMAL FATIGUE AND CATASTROPHIC THERMAL FAILURE

Printed circuit boards are comprised of dissimilar materials that expand at different rates of heating. Table 1 shows the thermal coefficients of expansion of materials commonly used in the fabrication of printed circuit boards. The differential expansion of mismatch must be accommodated by the various elements on the board. The increasing levels of packaging densities and board complexity dictate the need for designing a thermal environment that can accommodate the diverse components that are in close proximity to each other. [Ref. 4]

Table 1. MATERIALS USED IN PCB PRODUCTION

Chemical Name	Coefficient of Thermal Expansion $10^{-7}/°C$
Polyethylene	650
Bismaleimide	500
Polyamide imide	360
Polyarylate	625
Peek	400
Polyether amide	500
Polyimide	500
Polytetrafluorethylene	700-1200
Epoxy glass cloth	170 (x,y); 600 (z)
PBZT	90 (x,y); 200 (z)

Source: Reference 2.

As previously mentioned it is very important that the board designers have an understanding of the operating environment in which the board will be operating in order to incorporate into the design the tolerances that will allow the product to operate reliably.

Catastrophic thermal failure is defined as an immediate, thermally induced, total loss of electronic function in a specified component. This type of failure comes as a result of excessive temperature, or a thermal fracture. Catastrophic failure comes about as a result of many factors including the operating environment, equipment history, mechanical loading, and operational modes of the component. Although it is difficult to predict the temperature at which thermal failure may occur, it is possible to establish with the aid of thermal analysis the boundaries at which the board can be expected to operate reliably and within its useful operating life cycle. [Ref. 1,2]

III. HEAT TRANSFER

The degradation of the heat flow capabilities of a printed circuit board can lead to reliability problems due to excessive operating temperatures. It is imperative that designers incorporate into the board the capability to maintain temperatures within upper operational limits while operating in all possible environments in which the board will be exposed. [Ref. 2]

Heat transfer is defined as all energy flows that arise as a result of temperature differences. Because the components mounted on printed circuit boards and, indeed, the printed circuit boards themselves are not one hundred percent efficient, heat is generated. The primary modes of heat transfer are conduction and convection. Conductive modes include mechanical thermal contact and solid thermal interfaces between materials, such as copper, solder, or epoxy layers. Convective modes include natural and air-forced cooling, and forced liquid cooling. Radiation is also a factor; however, it is not as significant as conduction and convection at the temperatures in which printed circuit boards operate. [Ref. 5]

A. CONDUCTION

Conduction is the transfer by molecular motion of heat between one part of a body to another part of the same body or one body and another in physical contact. [Ref. 1]

For the case of conduction, the heat flow equation is the basis for understanding this behavior

$$Q = -kA\frac{\Delta T}{\Delta X} \tag{1}$$

where

Q = heat flow along the thermal gradient, $\frac{\Delta T}{\Delta X}$

A = the area through which the heat is flowing, m^2

k = the thermal conductivity of the material, $\frac{W}{m - {}^\circ C}$

$\frac{\Delta T}{\Delta X}$ = change in temperature per unit length, $\frac{°C}{m}$

Rearranging Equation (1) leads to

$$R_{th} = \frac{\Delta T}{Q} \tag{2}$$

which is the thermal resistance

$$R_{th} = \frac{\Delta X}{kA} \text{ in } \frac{°C}{W} \tag{3}$$

Equation (3) demonstrates that thermal resistance will increase with an increase in path length for heat flow, ΔX, with a decrease in area of heat flow, A, or change in conductivity k to a lower value.

1. General Equation of Heat Conduction

The general equation of heat conduction is

$$\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) + q_i = \rho C\frac{\partial T}{\partial t} \tag{4}$$

where

$\rho =$ density, $\frac{kg}{m^3}$

$C =$ specific heat, $\frac{J}{kg°C}$

$T =$ temperature, $°C$

$x, y,$ and $z =$ cartesian coordinates, m

$t =$ time, sec

$k =$ thermal conductivity, $\frac{W}{m - °C}$

$q =$ internal heat generation, $\frac{W}{m^3}$

Assuming k, C and ρ are independent of temperature, direction, and time, the resulting equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{5}$$

where

$\alpha =$ thermal diffusivity, $\frac{k}{\rho C}$, $\frac{m^2}{\sec}$

Several variations of the general conduction equation exist. Fourier's equation, which contains no heat sources is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha}\frac{\partial T}{\partial t} \tag{6}$$

Another variation, known as Poisson's equation, solves a system in which temperature is not time dependent

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} = 0 \tag{7}$$

The last variant of the general equation of conduction is intended for a system operating in a steady-state condition, and does not contain any heat sources. This equation, known as La Place's equation is

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \nabla^2 T = 0 \tag{8}$$

2. Single Plane Slab Modeling

For illustrative purposes, a single plane slab, with T_1 and T_2 being the face temperatures, is considered. If Equation (8) is limited to only one coordinate then

$$\frac{d^2 T}{dx^2} = 0 \tag{9}$$

If Equation (9) is integrated twice, and boundaries are established, then the temperature distribution across the slab can be expressed as follows

$$T = T_1 - \frac{x}{L}(T_1 - T_2) \tag{10}$$

If Equation (10) is substituted into Equation (1), then a solution for the heat flow across the single plane slab can be obtained

$$q = -kA\left[\frac{-(T_1 - T_2)}{L}\right] = \frac{kA}{L}(T_1 - T_2) \tag{11}$$

It is known that Ohm's law relates the voltage drop across a resistor with the current flow through the resistor, $V = IR$. Ohm's law is analogous to Equation (1), where the current flowing through a resistor is equivalent to heat flow, and the electrical resistance is equivalent to the thermal resistance expressed in Equation (2).

Ohm's law can be expressed as

$$I = \frac{V}{R} \tag{12}$$

where the analogy between Ohm's law and Equation (1) is

Current $I \Leftrightarrow$ Heat flow q

Potential $V \Leftrightarrow$ Temperature difference ΔT

Resistance $R \Leftrightarrow$ Thermal Resistance R

It can be observed that for the heat flow across a simple plane slab described by Equation (11), the thermal resistance is described by Equations (2) and (3).

B. CONVECTION

Convection is defined as the process by which thermal energy is transferred to or from a solid by a fluid flowing past it. If the movement of fluid is a result of a temperature differential, then the process is called free or natural convection. When natural convection is present, the movement of fluid can be accelerated by increasing the temperature differential. When a pressure differential is introduced to force the movement of the fluid by using a pump or a fan, the process is called forced convection. [Ref. 1,5]

Newton's law of cooling states that the heat flow through a body is proportional to the normal area and the temperature difference between the body and the surrounding fluid [Ref. 2]. To make it an equality, a proportionality constant, h , is introduced. The

proportionality constant is the surface heat transfer coefficient. Newton's law of cooling can be expressed as

$$q = hA(T_0 - T_f) \tag{13}$$

where

h is a proportionality factor known as the surface heat transfer coefficient

Comparing Newton's law of cooling with Fourier's law leads to an expression relating the surface heat transfer coefficient to thermal conductivity, the surface fluid temperature differences, and the wall temperature gradient of the fluid

$$h = \frac{q}{A\Delta T} = \frac{-\left(\frac{\partial T}{\partial y}\right)}{\Delta T} \tag{14}$$

Consequently, correlating heat transfer coefficients must be based on the dependence of h on the thermal conductivity of the fluid and on the ratio of the wall temperature gradient to the temperature difference. [Ref. 1]

1. Electrothermal Analog

For the case of convective heat transfer the thermal resistance is represented by

$$R = \frac{1}{hA} \tag{15}$$

Consequently, the total thermal resistance is now defined for a single slab and convective heat transfer on its two faces as follows

$$R = \frac{1}{h_1 A} + \frac{L}{kA} + \frac{1}{h_2 A} = \frac{1}{A}\left[\frac{1}{h_1} + \frac{L}{k} + \frac{1}{h_2}\right] \tag{16}$$

The heat transfer equation can now be represented as

$$q = \frac{\Delta T}{R} = \frac{(T_1 - T_2)}{\frac{1}{A}\left[\frac{1}{h_2} + \frac{L}{k} + \frac{1}{h_2}\right]} \tag{17}$$

C. RADIATION

The third mechanism for heat transfer is radiation in the form of electromagnetic waves. The rate at which a body radiates thermal energy is proportional to the area

of the body and to the fourth power of the absolute temperature [Ref. 5]. This result, found empirically by Josef Stefan in 1879, is written as

$$q = e\sigma A T^4 \tag{18}$$

where

$q =$ power radiated, W

$A =$ area, m^2

$e =$ emissivity of the body, a value between 0 and 1

$\sigma =$ Stefan-Boltzmann constant, $5.6703 \times 10^{-8} \dfrac{W}{m^2 \bullet K^4}$

When radiation falls on an opaque body, part of the radiation is reflected and part is absorbed. Light-colored bodies reflect most of the radiation, whereas dark bodies absorb most of it. [Ref. 5]

Materials employed in the manufacture of electronic components are classified as gray. Gray bodies are diffusely-reflecting opaque surfaces. These surfaces reflect equal amounts of energy over the thermal radiation spectrum in all directions. [Ref. 2]

1. Transformation of the General Radiation Equation

The use of the thermal radiation equation in analytical studies is a difficult task due to the fourth power relationship with temperatures. The complexity of the calculations involved with radiation dictate the need for the aid of a computer [Ref. 2]. The general equation for radiation interchange is

$$q = \sigma F_a F_e A (T_s^4 - T_r^4) \tag{19}$$

where, as before, σ is the Stefan-Boltzmann constant and

$F_a =$ shape factor accounting for source and receiver arrangement.

$A =$ area, m^2

$F_e =$ emissivity factor accounting for properties of the source and receiver.

$T_s =$ temperature of the source, $°K$

T_r = temperature of the receiver, $°K$

It must be observed that for the radiation case absolute temperature is the norm.

Radiation based heat transfer is represented by transforming the general radiation equation, Equation (19), into a form compatible with Fourier's law. Linearization of the general radiation equation is the method used to produce the desired result. The difference between the two fourth powers can be reduced to:

$$(T_s^4 - T_r^4) = (T_s^2 + T_r^2)(T_s^2 - T_r^2)$$

$$= (T_s^2 + T_r^2)(T_s + T_r)(T_s - T_r) \tag{20}$$

Inserting this into Equation (19) results in

$$q = \sigma F_a F_e A(T_s^2 + T_r^2)(T_s + T_r)(T_s - T_r) \tag{21}$$

A radiative heat transfer coefficient may be defined as

$$h_r = \sigma F_a F_e (T_s^2 + T_r^2)(T_s + T_r) \tag{22a}$$

or

$$h_r = \sigma F_a F_e (T_s^3 + T_s^2 T_r + T_s T_r^2 + T_r^3) \tag{22b}$$

Substitution of h_r into Equation (21) shows that radiative heat transfer can now be treated similarly to convection at the boundary [Ref. 6]. Thermal resistance in the case of radiation heat transfer can now be denoted as

$$R = \frac{1}{h_r A} \tag{23}$$

Now that all three methods of heat flow have been discussed, a graph depicting the electrothermal equivalent is shown in Figure 3. Here the plane slab has both radiation and convection on its two faces, and the nonlinearity of the radiation coefficient, h_r , leads to a more detailed analysis procedure.

When considering the calculation of heat transfer by radiation, it is usually necessary to approximate real material behavior with gray-body idealization. There is also a great deal of data required in order to model radiation for a real body not only

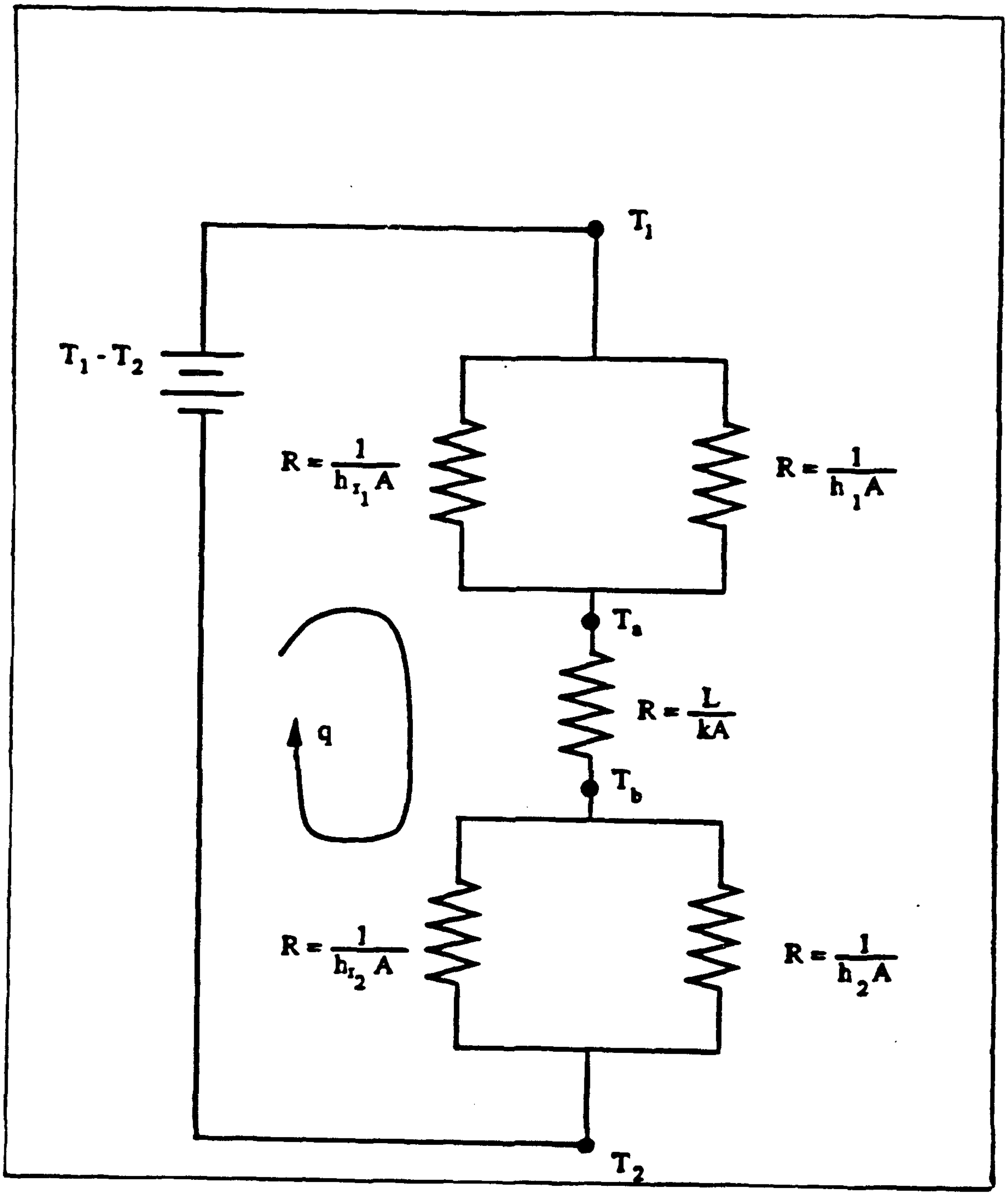

Figure 3. Radiation, convection, and conduction electrothermal equivalent.
Source: Reference 8.

because the surfaces are non-black, but also because the non-uniformity of the body's surfaces creates greater complexities in the collection and processing of data. [Ref. 7]

Basic problems, such as the one shown in Figure 3, defy a simple solution because of h_r. Therefore, a computer-aided approach becomes necessary in solving even elementary problems. Models exist for aiding the user in solving radiative heat transfer problems; however, their development is not within the boundaries of this thesis.

IV. FINITE DIFFERENCE ANALYSIS

Finite difference methods represent the most appropriate approach in determining the temperature distribution within an electronic component. By using a finite difference approach, complex problems involving difficult geometries, non linearities, or complicated boundary conditions can be successfully approached. The purpose of this section is to explain the fundamental concepts behind finite difference analysis and its use in solving differential equations. [Ref. 6]

A. FUNDAMENTALS OF FINITE DIFFERENCE ANALYSIS

The general equation for heat transfer by conduction, Equation (4), can be expressed in a three-dimensional variant as

$$\frac{\partial}{\partial x}(k\frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(k\frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(k\frac{\partial T}{\partial z}) + q_i = \rho c\frac{\partial T}{\partial t} \tag{24}$$

where all the equation components have been previously defined. In order to formulate a problem using Equation (24), it is first necessary to approximate the first and second derivatives.

1. First and Second Derivative Approximation

The derivative of a function at a point can be formulated by a finite difference approximation. It is first necessary to determine the slope of a line tangent to the point of interest on the plot of temperature as a function of x, y, or z. (Figure 4). In choosing x as the coordinate system, it is observed that the slope of the line (m) is defined as

$$\lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \tag{25}$$

The term of Equation (25), $[f(x_0 + \Delta x) - f(x_0)]/\Delta x$ is referred to as the difference quotient, and is the ratio of the change in the value of the function at x_0 and $x_0 + \Delta x$ to the change in x. The limit of the difference quotient is called the derivative of the function at x_0. [Ref. 8]

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} \tag{26}$$

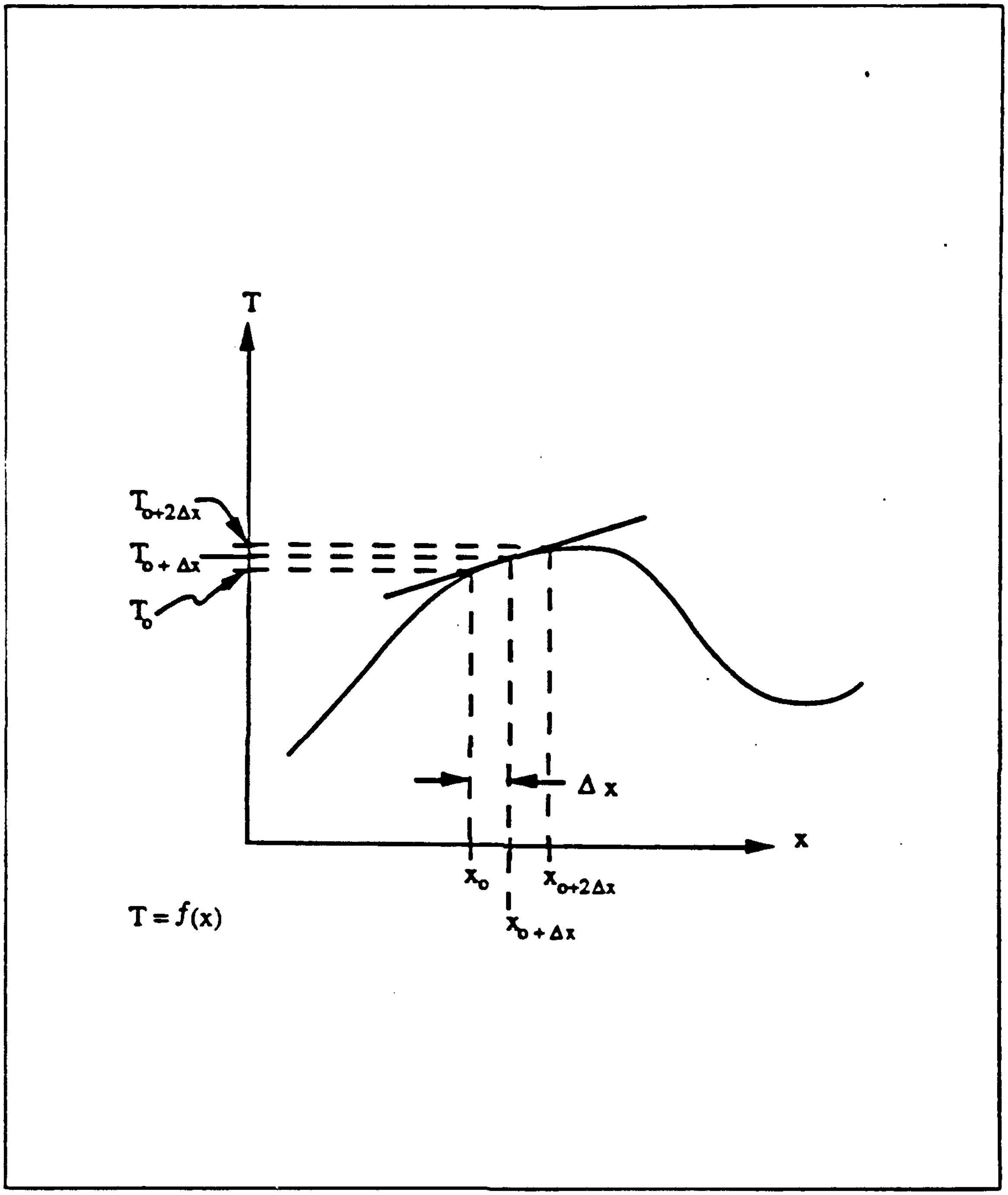

Figure 4. First and second derivative approximation.

Source: Reference 8.

By letting $f(x) = T(x)$, the derivative for this particular curve at the point x_0 can be derived

$$f'(x_0) = \lim_{\Delta x \to 0} \frac{T(x_0 + \Delta x) - T(x_0)}{\Delta x} \tag{27}$$

As $\Delta x \to 0$ Equation (27) becomes

$$f'(x_0) = \frac{dT}{dx} = \frac{T(x_0 + \Delta x) - T(x_0)}{\Delta x} \tag{28}$$

Thus, for Δx finite, but very small

$$f'(x_0) = \frac{\Delta T}{\Delta x} \tag{29}$$

The second derivative, using the definition of a derivative, will be the ratio of the change in value of the first derivative to the change in Δx [Ref. 8]. Again, looking at Figure 4, the second derivative is

$$f''(x_0) = \lim_{\Delta x \to 0} \frac{f'(x_0 + \Delta x) - f'(x_0)}{\Delta x} \tag{30}$$

$$f''(x_0) = \frac{d^2T}{dx^2} = \frac{T(x_0 + 2\Delta x) - 2T(x_0 + \Delta x) + T(x_0)}{\Delta x^2} \tag{31}$$

For both Equations (28) and (31), the approximations are valid as long as Δx is small.

Taylor's theorem states that a function can be approximated by a polynomial of the form

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2}(x - a)^2 + \dots + \frac{f^{(n)}(a)}{n!}(x - a)^n \tag{32}$$

where the polynomial is for the function expanded about $x = a$ [Ref. 8]. The Taylor series for the function $T(x)$ at $T(x_0 + 2\Delta x)$ is

$$T(x_0 + 2\Delta x) = T(x_0 + \Delta x) + \frac{dT(x_0 + \Delta x)}{dx}\Delta x + \frac{1}{2}\frac{d^2T(x_0 + \Delta x)}{dx^2}\Delta x^2 + \tag{33}$$

$$\frac{1}{6}\frac{d^3T(x_0 + \Delta x)}{dx^3}\Delta x^3 + \dots + \frac{1}{n!}\frac{d^nT(x_0 + \Delta x)}{dx^n}\Delta x^n$$

and for the function $T(x)$ at $T(x_0)$:

$$T(x_0) = T(x_0 + \Delta x) - \frac{dT(x_0 + \Delta x)}{dx}\Delta x + \frac{1}{2}\frac{d^2T(x_0 + \Delta x)}{dx^2}\Delta x^2 - \tag{34}$$

$$\frac{1}{6}\frac{d^3T(x_0 + \Delta x)}{dx^3}\Delta x^3 + \ldots + \frac{1}{n!}\frac{d^nT(x_0 + \Delta x)}{dx^n}\Delta x^n$$

If Δx is kept small, then the terms above second order become negligible. When Equations (33) and (34) are added, an expression for the second derivative can be obtained

$$T(x_0 + 2\Delta x) + T(x_0) = 2T(x_0 + \Delta x) + \frac{d^2T(x_0 + \Delta x)}{dx^2}\Delta x^2 \tag{35}$$

$$\frac{d^2T(x_0 + \Delta x)}{dx^2} = \frac{T(x_0 + 2\Delta x) - 2T(x_0 + \Delta x) + T(x_0)}{\Delta x^2} = f''(x_0) \tag{36}$$

As previously stated, the configuration to be analyzed can be subdivided into small, finite subvolumes considered to be isothermal. The centroid of each subvolume is called a node, and the node is representative of the entire subvolume. Nodes may are connected to their adjacent nodes through thermal resistances and the nodal analysis for the node temperatures can best be solved by means of a computer aided model [Ref. 8].

The first law of thermodynamics states that energy can neither be created nor destroyed, but can be transformed from one form to another. An energy balance may, therefore, be formed on the typical node shown in Figure 5. For the sake of simplicity, the environment is composed by a single node, node 5 (Figure 6). Close observation of node 5 shows that it is connected to nodes 2, 4, 6, 8, 14, and 100. If any energy is directly applied to or removed from this node, it would become part of the equation. The node equation for node 5, with rate of heat input q_i, becomes

$$q_2 + q_4 + q_6 + q_8 + q_{14} + q_{100} - q_l = 0 \tag{37}$$

where each one of the q values with a numerical subscript represents the rate of heat flow from node 5 to the node indicated by the subscript. [Ref. 1,6,8]

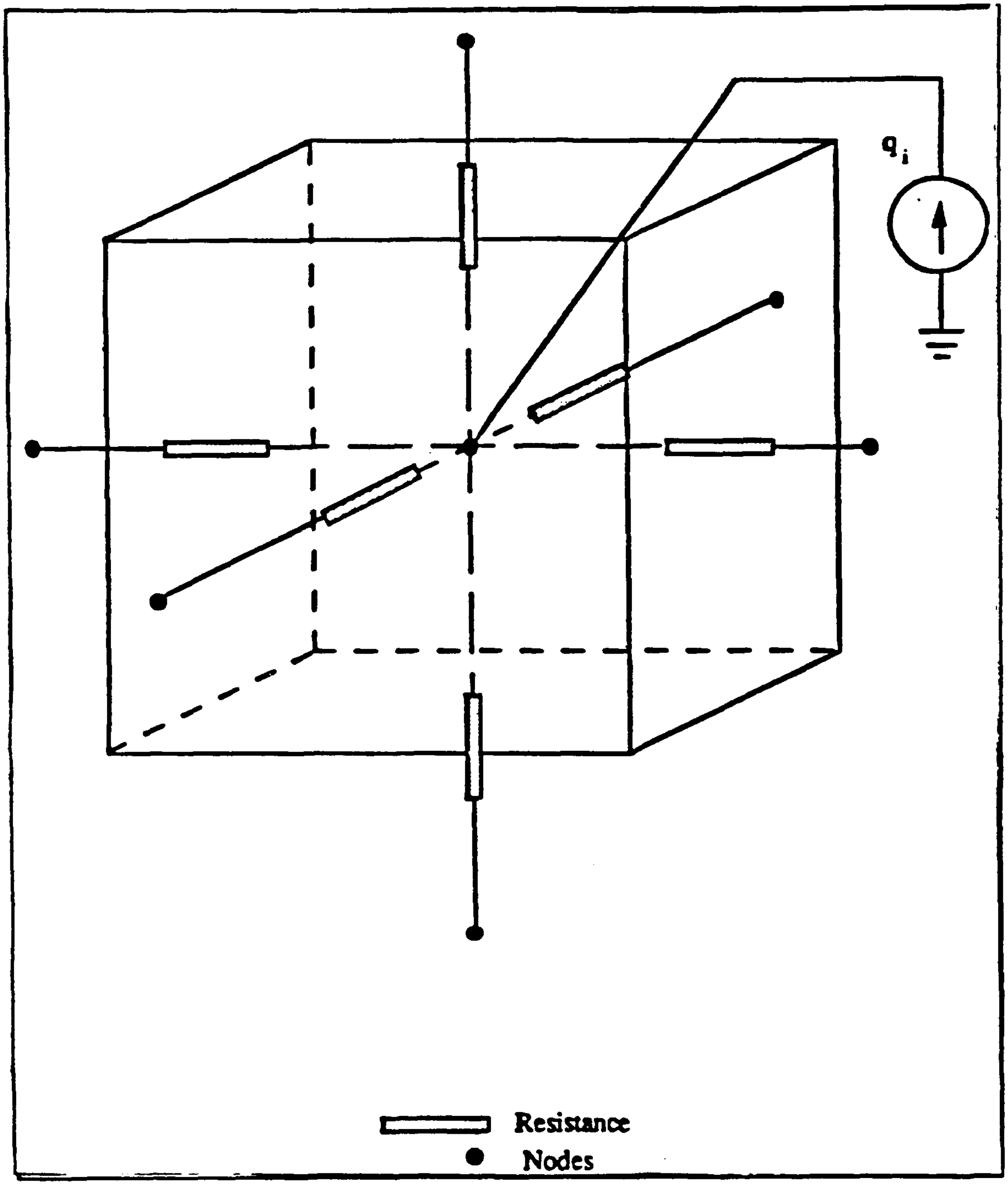

Figure 5. Graphical representation of a single node.
Source: Reference 6.

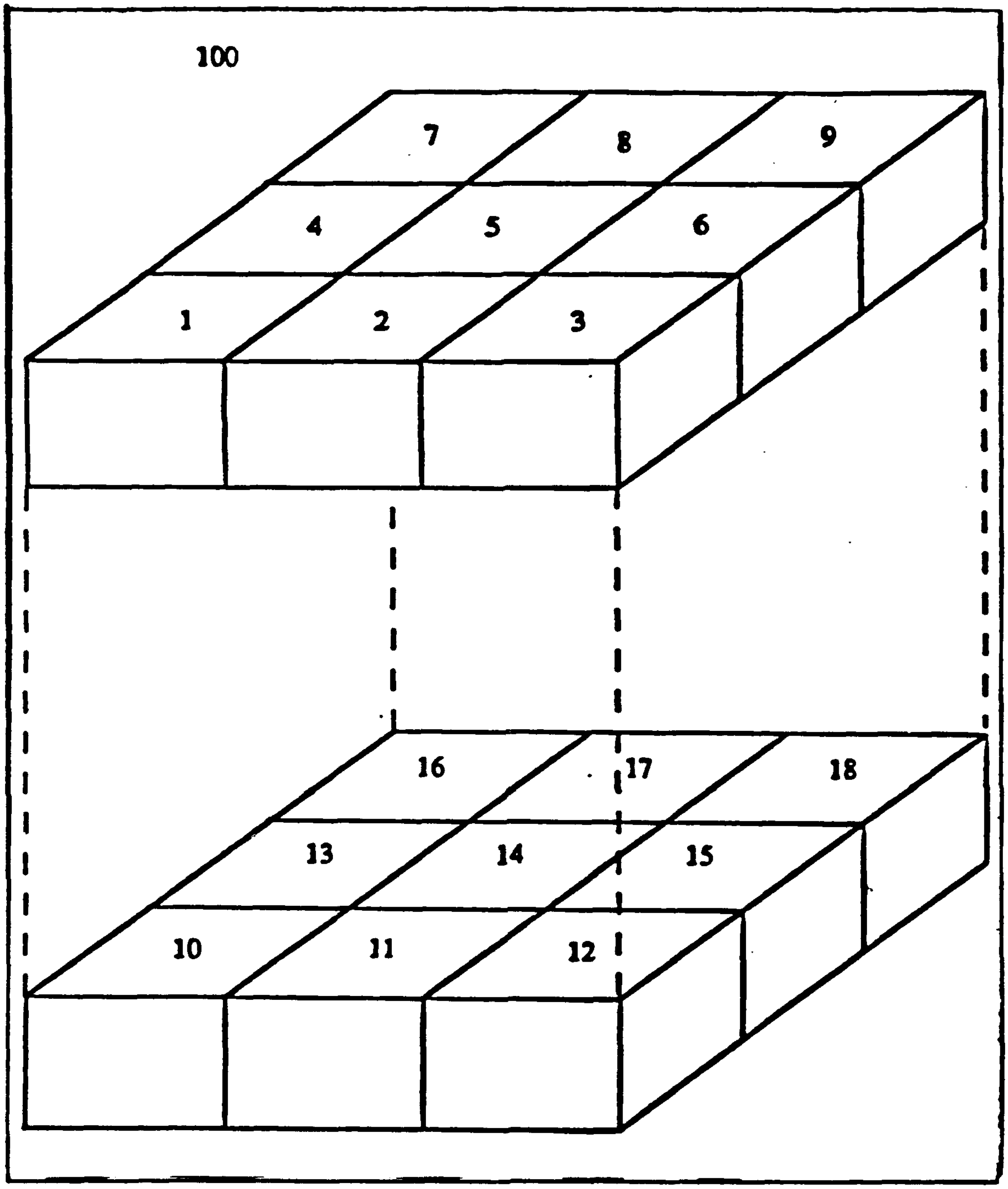

Figure 6. Node arrangement on typical printed circuit board.
Source: Reference 8.

Equation (37) does not display the node temperatures; however, $q = \Delta T/R$ and Equation (37) can be combined to produce

$$q_2 = \frac{kA}{L}\Delta T = \frac{k\Delta x\Delta z}{\Delta y}(T_5 - T_2) \tag{38a}$$

$$q_4 = \frac{kA}{L}\Delta T = \frac{k\Delta y\Delta z}{\Delta x}(T_5 - T_4) \tag{38b}$$

$$q_6 = \frac{kA}{L}\Delta T = \frac{k\Delta y\Delta z}{\Delta x}(T_5 - T_6) \tag{38c}$$

$$q_8 = \frac{kA}{L}\Delta T = \frac{k\Delta x\Delta z}{\Delta y}(T_5 - T_8) \tag{38d}$$

$$q_{14} = \frac{kA}{L}\Delta T = \frac{k\Delta x\Delta y}{\Delta z}(T_5 - T_{14}) \tag{38e}$$

$$q_{100} = \frac{kA}{L}\Delta T = \frac{k\Delta x\Delta y}{\frac{\Delta z}{2}}(T_5 - T_{100}) \tag{38f}$$

where T_{100} is on the top face of node 5 and, $\Delta x, \Delta y, \Delta z =$ the distance between nodes in the x, y, and z directions respectively. Assuming $\Delta x = \Delta y = \Delta z = 1$, then the node equation becomes

$$k(T_5 - T_2) + k(T_5 - T_4) + k(T_5 - T_6) + \tag{39}$$

$$k(T_5 - T_8) + k(T_5 - T_{14}) + 2k(T_5 - T_{100}) = q_i$$

or

$$-T_2 - T_4 + 7T_5 - T_6 - T_8 - T_{14} = \frac{q_i}{k} + 2T_{100} \tag{40}$$

where the terms on the right hand side of Equations (39) and (40) are known values. [Ref. 1,6,8]

A node equation is written for each node in the configuration, and in Figure 8, which contains 18 nodes, 18 node equations would be required to obtain the 18 unknown node temperatures. A larger problem having several hundred nodes would require prohibitive amounts of time to solve or even enter data for computer-aided thermal

analysis. That is why a computer-aided model builder capable of constructing the needed input data for thermal analysis, is so desirable. The thrust of this thesis is not to solve the equations, but to allow the user to model a printed circuit board to his or her specifications with minimal effort, and then produce a data file compatible for use in the thermal analyzer.

V. THE MODEL BUILDER

As previously indicated, the model builder currently in use in the thermal analysis software package requires time-consuming manual data entry. This thesis proposes that a model builder, PCB, can be incorporated into the software package. PCB is a menu-driven, user-friendly program which will assist the user in developing a thermal model of a printed circuit board with pre-defined geometries. PCB will generate a properly formatted ASCII output data file for use with the thermal analyzer. The current version of the thermal analyzer is designed to handle a maximum of 300 nodes [Ref. 9]. An updated version capable of handling the amount of nodes generated by the model builder is under development; however, abridged versions of the model builder have been successfully tested with the current version of the model builder. This chapter addresses many aspects of the model builder, including a terminal session demonstrating the design process used in modeling a circuit board, and the interface between the output data file and the thermal analyzer. Also addressed is the methodology used in validating the PCB model builder output data file. Finally, limitations and possible improvements to the model builder are discussed.

A. FEATURES

PCB is a menu driven interactive program providing the user with a variety of systematic choices progressing in a logical sequence that eventually will lead to the development of the model. Upon entering the program, the user will be provided with the option of viewing an optional overview. First-time users should consider reading the overview. The program is designed to operate with upper-case letters, therefore, it is suggested that the caps-lock key is pressed before starting the program.

Following the overview option the user will be asked to choose either SI or English units. For this choice as well all others, the program provides the user with the opportunity to correct any erroneous entries.

The third section queries the user for the number of copper layers desired in the design. The allowable number ranges from one to four layers. For every choice of number of copper layers, there are four possible board configurations the user can select. If one copper layer is selected, the program will display the following choices of aspect ratios:

1. 10 BY 36 PROVIDES A 1:3.6 RATIO WITH 720 NODES.

2. 10 BY 35 PROVIDES A 1:3.5 RATIO WITH 700 NODES.
3. 12 BY 25 PROVIDES A 1.2:2.5 RATIO WITH 600 NODES.
4. 15 BY 20 PROVIDES A 1.5:2.0 RATIO WITH 600 NODES.

When two copper layers are selected, the user will be provided with the following choices:

1. 10 BY 18 PROVIDES A 1:1.8 RATIO WITH 720 NODES.
2. 9 BY 20 PROVIDES A 1:2.11 RATIO WITH 720 NODES.
3. 8 BY 20 PROVIDES A 2:5 RATIO WITH 640 NODES.
4. 10 BY 15 PROVIDES A 1:1.5 RATIO WITH 600 NODES.

If three copper layers are selected, the user will have the following choices:

1. 5 BY 24 PROVIDES A 1:4.8 RATIO WITH 720 NODES.
2. 10 BY 10 PROVIDES A 1:1 RATIO WITH 600 NODES.
3. 12 BY 10 PROVIDES A 1.2:1 RATIO WITH 720 NODES.
4. 8 BY 15 PROVIDES A 1:1.875 RATIO WITH 720 NODES.

By selecting four copper layers, the program will provide the user the following options:

1. 5 BY 18 PROVIDES A 1:3.6 RATIO WITH 720 NODES.
2. 9 BY 10 PROVIDES A 1:1.11 RATIO WITH 720 NODES.
3. 8 BY 10 PROVIDES A 1:1.25 RATIO WITH 640 NODES.
4. 13 BY 6 PROVIDES A 1:1 RATIO WITH 624 NODES.

After selecting the desired aspect ratio, the user must specify the epoxy characteristics. The program will prompt the user to enter the epoxy layers length, width, thickness and thermal conductivity. An option available to the user will be the choice of a default epoxy thickness value, 0.0625 *in.*, (0.15875 *cm.*).

Once the epoxy characteristics have been entered, the program will then display the corresponding length and thickness for the copper layers selected. The length and width for the copper layers will be the same as those of the epoxy layers. The user will then be provided with the option of entering the thickness of each selected copper layer either by length or by weight. If input by length is selected, the units should be consistent with the system of measurement selected; however, if entry of thickness by weight is selected, then the entries should be in ounces. A copper layer weighing one *oz.* is equivalent to a copper layer 0.0014 *in.* thick with a surface area of 144 in^2 . The program adjusts

thelayer thickness to the surface area provided automatically. Entry of thickness by weight is a standard practice used in industry. [Ref. 2]

After entering the copper layer thickness the program will query the user for the thermal conductivity of copper. After this entry, the program will redisplay all information pertaining to the copper layers and will require the user to confirm all entries in order to proceed.

The following section requests the input of initial and ambient temperatures. PCB has six ambient temperatures.

After the temperatures have been entered the program requires the user to input the external heat sources. PCB provides for heat input into the upper copper layer. There are four methods of external heat input from which to choose. The first alternative allows for a total rate of heat applied to the upper surface. An entry for this choice would be divided by the number of nodes and distributed uniformly. The second choice provides for the entry of average heat per unit area. The third alternative gives the user the ability to enter heat in specifically designated nodes. The last option provides for no heat input and was developed to test compatibility with the thermal analyzer.

The following selection is the percent copper coverage for each copper layer. The program will prompt the user to enter the percentage of copper coverage for each layer in the printed circuit board.

Finally, the program requires a name and a title for the file that will be created. This file will be in the proper format for use with the thermal analyzer.

B. THE THERMAL ANALYZER INPUT DATA FILE

The model builder generates an ASCII data file from the physical characteristics of the printed circuit board provided by the user. In order for the input data file to be acceptable to the thermal analyzer, it must be in a specific format which is compatible to the thermal analyzer [Ref. 9]. Because each value and position of the output data file has a meaning to the thermal analyzer, and is not readily identifiable to the user, it is beneficial to describe each line and data set of the output data file and their relationship to PCB. Figure 7 shows a partial output data file.

Line one is the title line. It may be left blank or may contain up to 79 alphanumeric characters. The user-selected title appears at the top of the data file. [Ref. 6]

Line two is the problem data line. It has nine entries of which two are under user control, the number of nodes under consideration and the unit type. One entry, the number of constant temperatures, is preset at six for this specific model. The remaining

```
THIS IS A TEST OF THE PROGRAM
 640      6      0      0      0      0      0      0      1
   0      0      0
 750     50      6      2      4      6      0      0      0
   0
 .8500000  .6666670            12  .8000000  89.00000
     78.000      78.000      89.000      78.000      89.000      89.000
  7   7551        21      7521        91      7511       811      9991
  1.560        .624      .274      .137     2.787     2.787      .563
  7     11        31      7521       101      7511       821      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     21        41      7521       111      7511       831      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     31        51      7521       121      7511       841      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     41        61      7521       131      7511       851      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     51        71      7521       141      7511       861      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     61        81      7521       151      7511       871      9991
   .624        .624      .274      .137     2.787     2.787      .563
  7     71      7541      7521       161      7511       881      9991
   .624       1.560      .274      .137     2.787     2.787      .563
  7     81       101        11       171      7511       891      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7     91       111        21       181      7511       901      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    101       121        31       191      7511       911      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    111       131        41       201      7511       921      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    121       141        51       211      7511       931      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    131       151        61       221      7511       941      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    141       161        71       231      7511       951      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    151       171        81       241      7511       961      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    161       181        91       251      7511       971      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    171       191       101       261      7511       981      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    181       201       111       271      7511       991      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    191       211       121       281      7511      1001      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    201       221       131       291      7511      1011      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    211       231       141       301      7511      1021      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    221       241       151       311      7511      1031      9991
   .624        .624      .137      .137     2.787     2.787      .563
  7    231       251       161       321      7511      1041      9991
   .624        .624      .137      .137     2.787     2.787      .563
```

Figure 7. Sample PCB partial output data file.

entries have applications to models associated with heaters, unique exponents, secondary heat input, temperature coefficients and curves, and nodes controlling fast heat. These entries are not applicable to this model and are preset to zero. [Ref. 6]

Line three places a zero at three points and is beyond the user's control. Therefore, no further discussion is required. [Ref. 6]

Line four is the problem capability line. This line defines the maximum values for the entries in line two. The first entry is 750, which is the number of nodes for which the analysis is dimensioned. The number 750 is significant because the first constant temperature will be assigned to node number 751. The second entry is 50 which represents the largest possible number of constant temperatures in accordance with the analyzer dimension statement. The third entry is set to 6 and does not change. This entry is related to heaters and is not applicable to the model. The balance of the entries in line four represent a listing of data sets that are required for the particular analysis at hand. PCB uses three data sets that will be discussed in what follows. [Ref. 6]

The fifth line contains five values that relate to the accuracy level that the thermal analyzer will achieve. These entries are preset. The first value provides the level of accuracy between iterations. The accuracy level number is critical because too small a tolerance will cause the computer to run for excessive amounts of time, and too large a number will provide inaccurate results. The second value is the damping factor used between iterations in order to prevent temperature oscillations between iterations. The third number provides the maximum number of iterations. If erroneous data is entered, the computer will not run for excessive amounts of time. The fourth value is the convergence factor which adjusts the damping in order to close the critical value. The fifth entry is the initial temperature at which the iterative process begins. This value is supplied by the user. [Ref. 6]

Line six contains the temperature dependent coefficients, and is not used in this model. Line seven contains up to 50 constant temperature inputs. This model has six ambient temperatures entered by the user. [Ref. 6]

The following lines contain all pertinent information concerning the n-node equations. Each node requires two lines of data. Even-numbered lines are used for specifying the nodes that interact with the node in question and the modes by which this interaction takes place. For example, a line of the form:

```
6  7551  21  7521  111  7511  2411
```

is for node number one of aspect ratio selection one. The first entry indicates the number of connections to that node. The second entry is indicating that node 755, an ambient temperature, is connected to the node in question and the 1 indicates that the connection is conductive. The same procedure applies to the rest of the values on the line. If the entry is 9991 then the heat input is external. [Ref. 6]

Other data sets that appear relate to unique exponents, secondary heat, and temperature-dependent heat input curves. These data sets are not used in this model. [Ref. 6]

C. PCB MODEL BUILDER SAMPLE PROBLEM

The following terminal session is a typical example of the printed circuit board model builder, PCB, and its capabilities. Figure 8 depicts the printed circuit board modeled in the terminal session.

1. Printed Circuit Board Specifications

1. Unit system: British.
2. Copper layers in printed circuit board: three.
3. Aspect ratio desired: 12 by 10, with 720 nodes.
4. Epoxy layer length: 7.2 *in.*
5. Epoxy layer width: 4.2 *in.*
6. Epoxy layer thickness to default value: Yes (.0625 *in.*)
7. Epoxy layer thermal conductivity: 0.087 Btu/hr/°*F.*
8. Specify copper layer thickness by weight or length: length.
9. Thickness for the three copper layers, respectively: .20, .10, .11 *in.*
10. Copper thermal conductivity: 243.000 Btu/hr/°*F.*
11. Initial board temperature: 87 °*F.*
12. Upper surface ambient temperature: 86 °*F.*
13. Lower surface ambient temperature: 85 °*F.*
14. Right surface ambient temperature: 84 °*F.*
15. Left surface ambient temperature: 86 °*F.*
16. Front surface ambient temperature: 86 °*F.*
17. Rear surface ambient temperature: 86 °*F.*
18. Heat input to the upper copper layer: 300.23 Btu/hr/°*F.*
19. Percent copper coverage for copper layers, respectively: 87.3, 76.2, 65.7 %.
20. Name of output data file: LAYERS.

21. Header of the output data file: THIS IS THE OUTPUT OF THE THERMAL ANALYZER.

2. Terminal Session

Using the board specifications and Figure 8, the terminal session for the PCB program is as follows:

```
THIS PROGRAM WAS WRITTEN TO INTEGRATE WITH EXISTING
THERMAL ANALYSIS SOFTWARE AND TO REDUCE THE AMOUNT
OF TIME REQUIRED FOR DATA ENTRY.

WOULD YOU LIKE AND CVERVIEW OF THE PROGRAM PRIOR TO
BEGINNING? ENTER Y FOR YES AND N FOR NO: N


 ************************************************
 ************************************************
 **                                            **
 **      PLEASE SELECT UPPER CASE LETTERS      **
 **             PRIOR TO BEGINNING             **
 **                                            **
 ************************************************
 ************************************************


PRIOR TO ENTERING DATA INTO THIS PROGRAM ENSURE THAT
YOU HAVE A DRAWING OF YOUR DESIGN AND ALL PERTINENT
DATA.

              PRESS <ENTER> TO CONTINUE
```

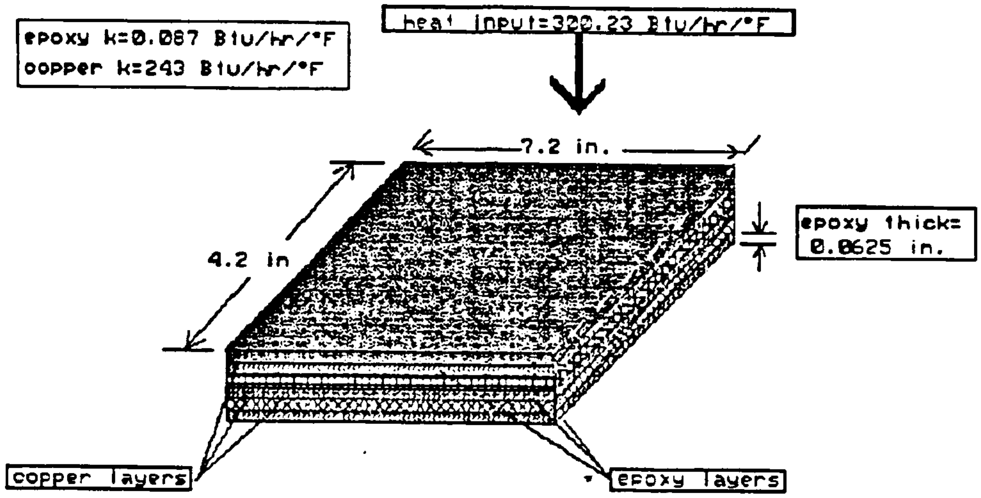

Figure 8. Printed circuit board modeled in the terminal session.

```
THIS PROGRAM IS CAPABLE OF OPERATIONS IN EITHER SI OR
ENGLISH UNITS. AFTER THE SELECTION OF THE UNITS, ALL
ENTRIES MUST BE COMPATIBLE. PLEASE MAKE YOUR SELECTION.

S FOR SI NOTATION
E FOR ENGLISH NOTATION:  E


YOU HAVE SELECTED ENGLISH NOTATION.

IS THIS THE DESIRED SELECTION? ENTER Y FOR YES AND N FOR NO:  Y


     THE PCB CAN HAVE UP TO FOUR COPPER LAYERS

     HOW MANY COPPER LAYERS DO YOU DESIRE?

     PICK A NUMBER 1 TO 4: 3


YOU SELECTED 3 COPPER LAYER(S) FOR THE PCB

IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FOR NO:   Y
```

```
YOU SELECTED 3 COPPER LAYERS, GIVING YOU THE FOLLOWING ALTERNATIVES:

1. 5 BY 24 PROVIDES A 1:4.8 RATIO WITH 720 NODES.
2. 10 BY 10 PROVIDES A 1:1 RATIO WITH 600 NODES.
3. 12 BY 10 PROVIDES A 1.2:1 RATIO WITH 720 NODES.
4. 8 BY 15 PROVIDES A 1:1.875 RATIO WITH 720 NODES.

PLEASE SELECT A NUMBER 1 THROUGH 4:    3



YOU SELECTED NUMBER 3 OF THE FOLLOWING 4 ALTERNATIVES:

1. 5 BY 24 PROVIDES A 1:4.8 RATIO WITH 720 NODES.
2. 10 BY 10 PROVIDES A 1:1 RATIO WITH 600 NODES.
3. 12 BY 10 PROVIDES A 1.2:1 RATIO WITH 720 NODES.
4. 8 BY 15 PROVIDES A 1:1.875 RATIO WITH 720 NODES.


IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FOR NO:   Y
```

```
*************************************************************************
**************************EPOXY CHARACTERISTICS***********************
*************************************************************************

 ALL ENTRIES ARE IN ENGLISH NOTATION


 ENTER EPOXY LAYER LENGTH (in): 7.2

 ENTER EPOXY LAYER WIDTH (in):  4.2


 THE EPOXY LAYER THICKNESS WILL DEFAULT TO 0.0625 in (0.15875 cm)
 DO YOU WANT TO CHANGE THE EPOXY THICKNESS? (Y OR N): N

 ENTER EPOXY LAYER THERMAL CONDUCTIVITY (Btu/hr/F  ):  0.087


    YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE EPOXY LAYER:


      1.) LENGTH:        7.2000 in
      2.) WIDTH:         4.2000 in
      3.) THICKNESS:      .0625 in
      4.) k:              .0870 Btu/hr/F


     DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR YES AND N FOR NO:   N
```

```
****************************************************************************
****************COPPER LAYER CHARACTERISTICS****************************
****************************************************************************

ALL ENTRIES ARE IN ENGLISH NOTATION

THE COPPER LAYER LENGTH IS THE SAME AS THE EPOXY LAYER:   7.2000 in

THE COPPER LAYER WIDTH IS THE SAME AS THE EPOXY LAYER:    4.2000 in

YOU SELECTED 3 COPPER LAYER(S) FOR THE PCB.
YOU NOW WILL BE ASKED TO ENTER THE COPPER LAYER THICKNESS

DO YOU WANT TO SPECIFY THICKNESS BY LENGTH OR WEIGHT?
ENTER L IF YOU WANT TO ENTER LENGTH, W IF YOU WANT TO ENTER WEIGHT: L

ENTER THE THICKNESS FOR LAYER 1 (in): .2

ENTER THE THICKNESS FOR LAYER 2 (in): .1

ENTER THE THICKNESS FOR LAYER 3 (in): .11

ENTER COPPER LAYER THERMAL CONDUCTIVITY (Btu/hr/F  ): 243

  YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE COPPER LAYER(S).

  1.) LENGTH:                 7.2000 in
  2.) WIDTH:                  4.2000 in
  3.) THICKNESS LAYER 1:       .2000 in
      THICKNESS LAYER 2:       .1000 in
      THICKNESS LAYER 3:       .1100 in
  4.) k:                    243.0000 Btu/hr/F
```

```
********************************************************************************
******************AMBIENT TEMPERATURE INPUTS*****************************
********************************************************************************


ENTER THE INITIAL BOARD TEMPERATURE (F): 87

ENTER THE UPPER SURFACE AMBIENT TEMPERATURE (F): 86

ENTER THE LOWER SURFACE AMBIENT TEMPERATURE (F): 85

ENTER THE RIGHT SURFACE AMBIENT TEMPERATURE (F): 84

ENTER THE LEFT SURFACE AMBIENT TEMPERATURE (F): 86

ENTER THE FRONT SURFACE AMBIENT TEMPERATURE (F): 86

ENTER THE REAR SURFACE AMBIENT TEMPERATURE (F): 86



YOU HAVE MADE THE FOLLOWING AMBIENT TEMPERATURE ENTRIES:


1. INITIAL BOARD TEMPERATURE:      87.000 F
2. UPPER AMBIENT TEMPERATURE:      86.000 F
3. LOWER AMBIENT TEMPERATURE:      85.000 F
4. RIGHT AMBIENT TEMPERATURE:      84.000 F
5. LEFT AMBIENT TEMPERATURE:       86.000 F
6. FRONT AMBIENT TEMPERATURE:      86.000 F
7. REAR AMBIENT TEMPERATURE:       86.000 F


DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR YES AND N FOR NO: N



DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR YES AND N FOR NO: N
```

****
******************************HEAT INPUTS********************************
****

HEAT INPUT TO THE PCB OCCURS ONLY ON THE UPPER COPPER LAYER. HEAT INPUT IS ACCOMPLISHED BY ONE OF THE FOLLOWING METHODS:

1. ENTER AS TOTAL HEAT APPLIED TO THE PCB
2. ENTER AS AVERAGE HEAT PER UNIT AREA
3. ENTER HEAT NODE BY NODE
4. NO HEAT INPUT

PLEASE SELECT NUMBER 1 THROUGH 4: 1

YOU HAVE SELECTED NUMER 1 OF FOUR ALTERNATIVES.
IS THIS THE DESIRED SELECTION? (Y OR N): Y

YOU HAVE SELECTED TO INPUT HEAT AS A TOTAL HEAT APPLIED TO THE SURFACE.

ENTER TOTAL HEAT APPLIED TO THE SURFACE (Btu/hr): 300.23

IS THIS THE CORRECT ENTRY? (Y OR N): Y

TOTAL HEAT PER NODE IS: 2.5019 Btu/hr

<PRESS ENTER TO CONTINUE>

ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LAYER 1: 87.3

ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LAYER 2: 76.2

ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LAYER 3: 65.7

```
YOU HAVE SELECTED:  87.30% COVERAGE, LAYER 1
                    76.20% COVERAGE, LAYER 2
                    65.70% COVERAGE, LAYER 3.

IS THIS YOUR DESIRED ENTRY?   (Y OR N):  Y
```

```
YOU SELECTED LAYERS FOR YOUR DATA FILE NAME

IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FOR NO:   Y
```

```
THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY INTO THE
EXISTING THERMAL ANALYZER, FURTHERMORE, THIS PROGRAM DOES
NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE
THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS
PROGRAM.  THE FILE NAME IS LIMITED TO SIX CHARACTERS, AND
SHOULD NOT HAVE ANY SPACES.

PLEASE ENTER THE DESIRED DATA FILE NAME:  LAYERS
```

```
YOU SELECTED LAYERS FOR YOUR DATA FILE NAME

IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FOR NO:   Y
```

ENTER THE DESIRED TITLE TO BE PLACED ON LINE
NUMBER ONE OF THE OUTPUT DATA FILE:

THIS IS THE OUTPUT OF THE THERMAL ANALYZER

DO YOU WISH TO CHANGE THE TITLE OF YOUR OUTPUT DATA FILE?
ENTER Y FOR YES AND N FOR NO: N

THE OUTPUT DATA HAS BEEN PLACED IN A FILE NAMED LAYERS

<PRESS ENTER TO CONTINUE>

3. The output data file

Figure 9 shows a partial listing of the model builder output computed in the terminal session. A detailed description of the output data file is provided in section B of this chapter.

4. Thermal analyzer output

The file obtained in the model builder produces an output for 720 nodes. In order to use this data file with the thermal analyzer the data file must first be shortened. In this particular case, the output file, LAYERS, shortened to eight nodes was entered to the thermal analysis program, TASS. Three files are produced by the thermal analysis program, TASS. The first two output files consist of detailed and summarized thermal analysis results, respectively. The third file is an error message summary in the event there is an error in the input data file. Figure 10 shows the detailed output file of the eight node thermal analysis.

D. PCB MODEL BUILDER VALIDATION

The output of the model builder was validated by following a three-step process. The first step in the validation process concerned format. In order to proceed with any other validation steps it was first necessary to ensure that all data fields were in accordance with the specifications set forth in the user's manual of the thermal analyzer [Ref. 9]. The thermal analyzer input data file consists of five lines and as many as seven input data sets. The thermal analyzer's manual model builder program, THANSS, was instrumental in verifying the accuracy of the model builder output data file format.

The second step entailed content. The output of the model builder was checked for accuracy and completeness. Thermal resistances in the output data files were checked and compared with manually calculated benchmark models. A benchmark model was developed for each copper layer configuration. Other information such as node numbers and node relationships were also compared with the benchmark model and verified for accuracy. To aid in the verification process, the manual model builder program, THANSS, part of the thermal analyzer software was used to verify node relationships [Ref. 9]. Other items checked for correctness were specification codes relating to units (British or SI), temperature inputs, and thermal analyzer default values.

The third and final step in the validation process required running the thermal analyzer program with the PCB output data file as the program input. Abridged versions of the model builder were successfully executed with the current version of the thermal analyzer (maximum node capability is 300 nodes). The results of the thermal analysis

```
THIS IS THE OUTPUT OF THE THERMAL ANALYZER
 720       6        0         0         0         0         0         0         1
  0        0        0
 750      50        6         2         4         6         0         0         0
 0
 .0500000  .6666670              12  .8000000  87.00000
     86.000        86.000        86.000       84.000        86.000        85.000
 7   7551          21         7521          131         7511         1211
 242.445       87.280      41.249       20.625        .701         .701          2.50
 7     11          31         7521          141         7511         1221
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     21          41         7521          151         7511         1231
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     31          51         7521          161         7511         1241
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     41          61         7521          171         7511         1251
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     51          71         7521          181         7511         1261
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     61          81         7521          191         7511         1271
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     71          91         7521          201         7511         1281
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     81         101         7521          211         7511         1291
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7     91         111         7521          221         7511         1301
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7    101         121         7521          231         7511         1311
 87.280        87.280      41.249       20.625        .701         .701          2.50
 7    111        7541         7521          241         7511         1321
 87.280       242.445      41.249       20.625        .701         .701          2.50
 7    121         141           11          251         7511         1331
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    131         151           21          261         7511         1341
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    141         161           31          271         7511         1351
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    151         171           41          281         7511         1361
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    161         181           51          291         7511         1371
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    171         191           61          301         7511         1381
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    181         201           71          311         7511         1391
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    191         211           81          321         7511         1401
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    201         221           91          331         7511         1411
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    211         231          101          341         7511         1421
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    221         241          111          351         7511         1431
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    231         251          121          361         7511         1441
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    241         261          131          371         7511         1451
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    251         271          141          381         7511         1461
 87.280        87.280      20.625       20.625        .701         .701          2.50
 7    261         281          151          391         7511         1471
```

Figure 9. Partial output data file of PCB terminal session.

were compared with the results obtained using the analyzer's manual model builder, THANSS, and checked for discrepancies.

```
                                                              Page No.   1
  THIS IS THE OUTPUT OF THE THERMAL ANALYZER
 Temperatures, deg


 Node            1          2          3          4          5          6
 1st Tmp   85.00      85.00      85.00      85.00      85.00      85.00

 Node            7          8
 1st Tmp   85.00      85.00


 Node            1          2          3          4          5          6
 New Tmp   85.00      85.00      85.00      85.00      85.00      85.00
 New-Old   .7998      .7998      .7998      .7998      .7997      .7997

 Node            7          8
 New Tmp   85.00      85.00
 New-Old   .7997      .7998


 Node            1          2          3          4          5          6
 New Tmp   85.96      85.96      85.96      85.96      85.96      85.96
 New-Old   .1600      .1600      .1600      .1600      .1600      .1600

 Node            7          8
 New Tmp   85.96      85.96
 New-Old   .1600      .1600


 Node            1          2          3          4          5          6
 New Tmp   86.00      86.00      86.00      86.00      86.00      86.00
 New-Old   .4034E-01  .4036E-01  .4038E-01  .4041E-01  .4043E-01  .4046E-01

 Node            7          8
 New Tmp   86.00,     86.00
 New-Old   .4045E-01  .4044E-01

                                                              Page No.   1
  THIS IS THE OUTPUT OF THE THERMAL ANALYZER
 TEMPERATURES, DEGF
   1   100.78   2   104.26   3   105.30   4   105.96   5   106.06   6   106.03
   7   105.60   8   104.27
```

Figure 10. Output data file of TASS thermal analyzer.

E. MODEL BUILDER LIMITATIONS AND POSSIBLE IMPROVEMENTS

Although the model builder provides the user with a significant advantage in terms of time and effort necessary to model a printed circuit board, there are at present certain limitations to the program. There is also significant potential for improvements that could enhance the capabilities of the current version of the model builder.

1. PCB Model Builder Limitations

1. The model builder is restricted to a maximum of four copper layers due to conventional personal computer operating system limitations. Current versions of MS-DOS ® do not allow the user to access extended memory. Access to extended memory would allow for a larger copper layer capability for the model.
2. The model builder will not run when using MS-DOS ®, version 4.01. When running PCB on a computer operating with MS-DOS ®, version 4.01, the computer will display an "insufficient memory" message. Writing more compact code for the model builder could overcome this problem. The program will run on any other version of MS-DOS ®.
3. The model builder does not account for convective and radiative heat transfer.
4. The model builder only accepts heat inputs from the top surface of the printed circuit board.

2. Possible Improvements to the Thermal Model Builder.

1. The present version of the model builder requires extensive amounts of FORTRAN code. An updated model builder could be designed using a more compact and efficient higher level language, such as C++. Smaller, more efficient source code would allow for incorporating more copper layers into the model builder.
2. The menu driven queries used in the thermal analyzer could be improved by incorporating assembly language routines enabling the use of a mouse. Attempts were made to introduce the use of mouse driven menus; however, memory handling limitations in the Microsoft ® linker, version 3.55, prevented the use of a mouse.
3. Allowing for the injection of heat not only from the top surface of the printed circuit board, but also from other external surfaces (to include the heat being transferred through the wedges supporting the printed circuit board), is another potential improvement.
4. Adding graphics capabilities to the model builder would enable the user to model the printed circuit board without the need for preliminary drawings or sketches; however, adding this feature would cause the model builder executable code to become excessively large.

VI. CONCLUSIONS

The purpose of this thesis is to develop a printed circuit board thermal analysis model builder that will effectively interact with thermal analysis software used by the Naval Postgraduate School. [Ref. 9]

The current version of the model builder enables the user to choose from sixteen possible printed circuit board configurations, ranging from one to four copper layers. Other features of the model builder include the ability to set the physical characteristics of the board including dimensions, thermal conductivities, percent copper coverage, board temperatures, and heat input modes.

The main goal of the model builder is to enable the user to model a printed circuit board with minimum effort, and produce a thermal analysis input data file. Using the model builder, PCB, relieves the user of the tedious, time-consuming manual data entry required of the thermal analyzer's current model builder, THANSS. [Ref. 9]

The expansion possibilities for the model builder are significant. Adding more copper layers and nodes would enable the user to model more sophisticated printed circuit boards. Other potential features include the addition of graphics capabilities and the introduction of user-friendly peripherals, such as mouse drivers or digitazing pads. As the level of complexity of future model builders increases, it will become necessary to import the source code from FORTRAN compilers limited to 640 Kilobytes of memory, to 32 bit FORTRAN compilers allowing use of all RAM. The source code for PCB contains over 5000 lines of code, effectively reaching the memory limits of both the compiler and personal computer.

The possible applications for the model builder are substantial. As the level of complexity of printed circuit boards increases, there is a real need for a tool that allows the designer to efficiently perform thermal analysis of printed circuit boards during the design process. The model builder employed must allow the designer the necessary flexibility needed to model a board that meets the desired design criteria. The PCB model builder is the first step in creating a tool that enables the designer to effectively build a model that meets specified requirements.

APPENDIX PROGRAM LISTING FOR PCB MODEL BUILDER

```
$LARGE
C
C
C     TITLE:      MODEL BUILDER -- MAIN PROGRAM
C     AUTHOR:     LT STEVE GLASER
C     DATE:       09 JUL 1991
C     COMPILER:   MICROSOFT VERSION 4.01
C     LINKER:     MICROSOFT VERSION 3.55
C
C     DEFINE REAL VARIABLES
C

      REAL EL,EW,DELE,UKE,CL,CW,T1L1,T2L1,T2L2,T3L1,T3L2,T3L3,T4L1,T4L2,
     +T4L3,T4L4,T1W1,T2W1,T2W2,T3W1,T3W2,T3W3,T4W1,T4W2,T4W3,T4W4,AREA,U
     +N,VOL1,VOL2,VOL3,VOL4,UKC,IBT,UPRT,LWRT,RT,LT,FT,BT,THEAT,THPN,AHE
     +AT,NHEAT,DELX,DELY

      COMMON IBT,UPRT,LWRT,RT,LT,FT,BT,UT,ANSWER

      INTEGER NWIDE,NDEEP,NPL,NUM,NUMA,CH,H
C
C
C     DEFINE ALL ONE CHARACTER VARIABLES
C
C
C
      CHARACTER*1 SELECT,ANS,ANSL,OVR,ANSWER,ANSN,ANSA,LAY1,LAY2,LAY3,LA
     +Y4,ANSE,ANSEL,ANSEW,ANSET,ANSEK,ANSQ,WORL,ANSE1,ANST1,ANSK1,ANSQ1,
     +SPEVAL,ANSTEM,ANSTEB,ANSTEU,ANSTEL,ANSTER,ANSLFT,ANSFRT,ANSBCK
C
C
C     CHARACTER VARIABLES OF MORE THAN ONE POSITION
C
C

      CHARACTER UK*11,UT*1,UH*13,UAH*6,UL*2,UW*2
C
C
C     DEFINE MATRICES
C
       REAL HEAT(100,100),COEF(1000,12)
C
C
C
       INTEGER IH(1000),JH(1000),NCON(1000,12)
C
C
C     PROVIDE THE USER WITH AN INTRODUCTORY STATEMENT.
C
C
88888 CALL CLS
```

```
      WRITE(*,7001)
7001  FORMAT(///////,'                                                    ',/,
     +'     THIS PROGRAM WAS WRITTEN TO INTEGRATE WITH EXISTING   ',/,
     +'     THERMAL ANALYSIS SOFTWARE AND TO REDUCE THE AMOUNT    ',/,
     +'     OF TIME REQUIRED FOR DATA ENTRY.',//,
     +'     WOULD YOU LIKE AND OVERVIEW OF THE PROGRAM PRIOR TO   ',/,
     +'     BEGINNING? ENTER Y FOR YES AND N FOR NO:    ',2X, )
      READ(*,7002) OVR
7002  FORMAT(A1)
C
C
C
C
C    VARIABLE, CONSTANT, AND STRING DEFINITION
C         PHYSICAL CHARACTERISTICS
C           CL,EL - EPOXY AND COPPER LENGTHS
C           CW,EW - EPOXY AND COPPER WIDTHS
C           UKE,UKC - EPOXY AND COPPER THERMAL CONDUCTIVITIES
C           DELE, T1L1 THROUGH T4L4 - THICKNESS OF EPOXY AND COPPER LAYERS
C           DELX - CL/NDEEP
C           DELY - CW/NWIDE
C           NPL - NUMBER OF NODES PER LAYER
C           NWIDE - NUMBER OF NODES WIDE
C           NDEEP - NUMBER OF NODES DEEP
C           UL - UNITS OF LENGTH (SI OR ENGLISH)
C           UK - UNITS OF THERMAL CONDUCTIVITY (SI OR ENGLISH)
C
C    INITIAL AND AMBIENT TEMPERATURES
C
C           IT -  INITIAL BOARD TEMPERATURE
C           LT -  LEFT SIDE AMBIENT TEMPERATURE
C           RT -  RIGHT SIDE AMBIENT TEMPERATURE
C           FT -  FRONT AMBIENT TEMPERATURE
C           BT -  BACK AMBIENT TEMPERATURE
C           UPRT - UPPER AMBIENT TEMPERATURE
C           LWRT - LOWER AMBIENT TEMPERATURE
C           UT - UNITS OF TEMPERATURE (CENTIGRADE OR FAHRENHEIT)
C
C    HEAT INPUT
C
C           THEAT - TOTAL INJECTED HEAT
C           THPN  - TOTAL HEAT PER NODE
C           AHEAT - AVERAGE HEAT OVER A GIVEN SURFACE
C           NHEAT - NODE PER HEAT INJECTED NODE PER NODE
C           UH, UAH - UNITS OF HEAT (SI OR ENGLISH)
C           NUM, NUMA, CH, H - DUMMY VARIABLES
C           TOTNOD, NN, NC, N - VARIABLES USED TO ALLOW NODAL HEAT INPUT
C           IH, JH - VECTORS USED TO CORRELATE NODE NUMBER WITH
C                    MATRIX POSITION
C           HEAT - MATRIX USED TO HOLD HEAT INPUTS
C
C
C    COEFFICIENT DEFINITIONS     Y IMPLIES WIDTH, AND X IMPLIES DEPTH
C                                Z IMPLIES HEIGHT
C                                C IMPLIES COPPER, E IMPLIES EPOXY
C
```

```
C           EYLR - PROVIDES COEFFICIENT FOR LEFT OR RIGHT EDGE NODES
C                  TO THE EXTERNAL NODE, EYLR IMPLIES EPOXY.
C           CYLR - SAME AS ABOVE, COPPER LAYER
C
C           EXFB - PROVIDES COEFFICIENT FOR FRONT AND BACK EDGE NODES
C                  TO THE EXTERNAL NODE.
C           CXFB - SAME AS ABOVE FOR COPPER LAYER
C
C           EYY - INTERNAL COEFFICIENT IN THE Y DIRECTION, EPOXY
C           CYY - INTERNAL COEFFICIENT IN THE Y DIRECTION, COPPER
C
C           EXX - INTERNAL COEFFICIENT IN THE X DIRECTION, EPOXY
C           CXX - INTERNAL COEFFICIENT IN THE X DIRECTION, COPPER
C
C           SZE - COEFFICIENT FOR EPOXY TO UPPER EXTERNAL USE
C           EZC - COEFFICIENT FOR COPPER TO EPOXY
C           PCC - PERCENT COPPER COVERAGE
C
C           ALL ONE CHARACTER STRINGS ARE SIMPLE 'YES' OR 'NO' TYPE
C           VARIABLES.
C
C     CALL INTRODUCTION
C
C
      IF (OVR.EQ.'Y') THEN
      CALL INTRO
      ELSEIF (OVR.EQ.'N') THEN
      GOTO 899
      ELSE
      GOTO 88888
      ENDIF
C
C     SUPPLY A LIST OF ACCEPTABLE LAYER LEVELS
C
C     BEGIN PROGRAM
C
C
899   CALL CLS
      WRITE(*,800)
800   FORMAT(///,'          ************************************************
     +****',/,
     +'         ************************************************',/,
     +'         **                                            **',/,
     +'         **      PLEASE SELECT UPPER CASE LETTERS      **',/,
     +'         **             PRIOR TO BEGINNING             **',/,
     +'         **                                            **',/,
     +'         ************************************************',/,
     +'         ************************************************')
C
      WRITE(*,801)
801   FORMAT(//,'                                                 ',/,
     +'        PRIOR TO ENTERING DATA INTO THIS PROGRAM ENSURE THAT ',/,
     +'        YOU HAVE A DRAWING OF YOUR DESIGN AND ALL PERTINENT  ',/,
     +'        DATA.                                                ',///,
     +'                      PRESS <ENTER> TO CONTINUE           ',2X, )
```

```
      READ(*,802)ANSWER
802   FORMAT(A1)
C
C
C
      CALL CLS
357   WRITE(*,367)
367   FORMAT(///////,'       THIS PROGRAM IS CAPABLE OF OPERATIONS IN EI
     +THER SI OR ',/,
     +'       ENGLISH UNITS. AFTER THE SELECTION OF THE UNITS, ALL ',/,
     +'       ENTRIES MUST BE COMPATIBLE. PLEASE MAKE YOUR SELECTION.'/
     +/,'       S FOR SI NOTATION',/,
     +'       E FOR ENGLISH NOTATION:   ',2X, )
      READ(*,368) ANSN
368   FORMAT(A1)
C     CHECK FOR CORRECT UNIT SELECTION
C
3333  IF(ANSN.EQ.'S') THEN
      WRITE(*,369)
369   FORMAT(////,'       YOU HAVE SELECTED SI NOTATION.')
      ELSEIF (ANSN.EQ.'E') THEN
      WRITE(*,370)
370   FORMAT(////,'       YOU HAVE SELECTED ENGLISH NOTATION.')
      ELSE
         CALL CLS
         GOTO 357
      ENDIF
C
3711  WRITE(*,3710)
3710  FORMAT(//,'       IS THIS THE DESIRED SELECTION? ENTER Y FOR YES A
     +ND N FOR NO:   ', )
      READ(*,372) ANSA
372   FORMAT(A1)
C
      IF(ANSA.EQ.'Y') THEN
      GOTO 378
      ELSEIF(ANSA.EQ.'N') THEN
      CALL CLS
      GOTO 357
      ELSE
      CALL CLS
      GOTO 3333
378   ENDIF
C
C
C
C
7     CALL CLS
      SPEVAL='A'
      WRITE(*,301)
301   FORMAT(//////,'     THE PCB CAN HAVE UP TO FOUR COPPER LAYERS')
      WRITE(*,302)
302   FORMAT(//,'     HOW MANY COPPER LAYERS DO YOU DESIRE?')
      WRITE(*,303)
303   FORMAT(/,'     PICK A NUMBER 1 TO 4:  ',2X, )
```

```
      READ(*,304)SELECT
304   FORMAT(A1)
C
C
      CALL CLS
      IF (SELECT.NE.'1'.AND.SELECT.NE.'2'.AND.SELECT.NE.'3'.AND.
     +SELECT.NE.'4') THEN

      GOTO 7

      ELSE
10      WRITE(*,305)SELECT
305     FORMAT(//////,'  YOU SELECTED ',A1,' COPPER LAYER(S) FOR THE PCB
     +',//)
C
      ENDIF
C
      WRITE(*,306)
306   FORMAT('  IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FO
     +R NO:    ',2X, )
      READ(*,307)ANS
307   FORMAT(A1)
C
C
      IF(ANS.EQ.'N') THEN
        GOTO 7
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'Y') THEN
        GOTO 9
      ELSE
        CALL CLS
        GOTO 10
9     ENDIF
C
C
      CALL CLS
C
C
C
      IF(SELECT.EQ.'1') THEN
        GOTO 123
      ELSE
        CONTINUE
      ENDIF
      IF(SELECT.EQ.'2')THEN
        GOTO 124
      ELSE
        CONTINUE
      ENDIF
      IF(SELECT.EQ.'3')THEN
        GOTO 125
      ELSE
        CONTINUE
      ENDIF
```

```
      IF(SELECT.EQ.'4')THEN
        GOTO 126
      ELSE
        CONTINUE
      ENDIF
        GOTO 7
C
C
11    CALL CLS
123   WRITE(*,308)
308   FORMAT(//////,'  YOU SELECTED 1 COPPER LAYER, GIVING YOU THE FOLLO
     +WING ALTERNATIVES:')
      WRITE(*,309)
309   FORMAT(//,'  1. 10 BY 36 PROVIDES A 1:3.6 RATIO WITH 720 NODES.
     +',/,'  2. 10 BY 35 PROVIDES A 1:3.5 RATIO WITH 700 NODES.',/,
     +'  3. 12 BY 25 PROVIDES A 1.2:2.5 RATIO WITH 600 NODES.',/,
     +'  4. 15 BY 20 PROVIDES A 1.5:2.0 RATIO WITH 600 NODES.',//,
     +'  PLEASE SELECT A NUMBER 1 THROUGH 4:     ',2X, )
      READ(*,310)LAY1
310   FORMAT(A1)
      IF (LAY1.NE.'1'.AND.LAY1.NE.'2'.AND.LAY1.NE.
     +'3'.AND.LAY1.NE.'4') THEN
      GOTO 11
      ELSE
3334  CALL CLS
      WRITE(*,311)LAY1
311   FORMAT(//////,'  YOU SELECTED NUMBER 'A1' OF THE FOLLOWING 4 ALTER
     +NATIVES:',///,
     +'  1. 10 BY 36 PROVIDES A 1:3.6 RATIO WITH 720 NODES.',/,
     +'  2. 10 BY 35 PROVIDES A 1:3.5 RATIO WITH 700 NODES.',/,
     +'  3. 12 BY 25 PROVIDES A 1.2:2.5 RATIO WITH 600 NODES.',/,
     +'  4. 15 BY 20 PROVIDES A 1.5:2.0 RATIO WITH 600 NODES.',///)
      ENDIF
13    WRITE(*,312)
312   FORMAT('  IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FO
     +R NO:    ',2X, )
      READ(*,313)ANSL
313   FORMAT(A1)
C
C
      IF(ANSL.EQ.'N') THEN
        GOTO 11
      ELSE
        CONTINUE
      ENDIF
      IF(ANSL.EQ.'Y') THEN
        GOTO 12
      ELSE
        GOTO 3334
12    ENDIF
      GOTO 1234
14    CALL CLS
124   WRITE(*,314)
314   FORMAT(//////,'  YOU SELECTED 2 COPPER LAYERS, GIVING YOU THE FOLL
     +OWING ALTERNATIVES:')
```

```
      WRITE(*,315)
315   FORMAT(//,'  1. 10 BY 18 PROVIDES A 1:1.8 RATIO WITH 720 NODES.
     +',/,'  2. 9 BY 20 PROVIDES A 1:2.11 RATIO WITH 720 NODES.',/,
     +'  3. 8 BY 20 PROVIDES A 2:5 RATIO WITH 640 NODES.',/,
     +'  4. 10 BY 15 PROVIDES A 1:1.5 RATIO WITH 600 NODES.',//,
     +'  PLEASE SELECT A NUMBER 1 THROUGH 4:     ',2X, )
      READ(*,316)LAY2
316   FORMAT(A1)
      IF (LAY2.NE.'1'.AND.LAY2.NE.'2'.AND.LAY2.NE.
     +'3'.AND.LAY2.NE.'4') THEN
      GOTO 14
      ELSE
3335  CALL CLS
      WRITE(*,317)LAY2
317   FORMAT(//////,'  YOU SELECTED NUMBER 'A1' OF THE FOLLOWING 4 ALTER
     +NATIVES:',///,
     +'  1. 10 BY 18 PROVIDES A 1:1.8 RATIO WITH 720 NODES.',/,
     +'  2. 9 BY 20 PROVIDES A 1:2.11 RATIO WITH 720 NODES.',/,
     +'  3.  8 BY 20 PROVIDES A 2:5 RATIO WITH 640 NODES.',/,
     +'  4. 10 BY 15 PROVIDES A 1:1.5 RATIO WITH 600 NODES.',///)
      ENDIF
15    WRITE(*,318)
318   FORMAT('  IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FO
     +R NO:    ',2X, )
      READ(*,319)ANSL
319   FORMAT(A1)
C
C
      IF(ANSL.EQ.'N') THEN
        GOTO 14
      ELSE
        CONTINUE
      ENDIF
      IF(ANSL.EQ.'Y') THEN
        GOTO 16
      ELSE
        GOTO 3335
16    ENDIF
      GOTO 1235

17    CALL CLS
125   WRITE(*,320)
320   FORMAT(//////,'  YOU SELECTED 3 COPPER LAYERS, GIVING YOU THE FOLL
     +OWING ALTERNATIVES:')
      WRITE(*,321)
321   FORMAT(//,'  1. 5 BY 24 PROVIDES A 1:4.8 RATIO WITH 720 NODES.
     +',/,'  2. 10 BY 10 PROVIDES A 1:1 RATIO WITH 600 NODES.',/,
     +'  3. 12 BY 10 PROVIDES A 1.2:1 RATIO WITH 720 NODES.',/,
     +'  4. 8 BY 15 PROVIDES A 1:1.875 RATIO WITH 720 NODES.',//,
     +'  PLEASE SELECT A NUMBER 1 THROUGH 4:     ',2X, )
      READ(*,322)LAY3
322   FORMAT(A1)
      IF (LAY3.NE.'1'.AND.LAY3.NE.'2'.AND.LAY3.NE.
     +'3'.AND.LAY3.NE.'4') THEN
      GOTO 17
```

```
      ELSE
3336  CALL CLS
      WRITE(*,323)LAY3
323   FORMAT(//////,'  YOU SELECTED NUMBER 'A1' OF THE FOLLOWING 4 ALTER
     +NATIVES:',///,
     +'   1.  5 BY 24 PROVIDES A 1:4.8 RATIO WITH 720 NODES.',/,
     +'   2.  10 BY 10 PROVIDES A 1:1 RATIO WITH 600 NODES.',/,
     +'   3.  12 BY 10 PROVIDES A 1.2:1 RATIO WITH 720 NODES.',/,
     +'   4.  8 BY 15 PROVIDES A 1:1.875 RATIO WITH 720 NODES.',///)
      ENDIF
19    WRITE(*,324)
324   FORMAT('  IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FO
     +R NO:    ',2X, )
      READ(*,325)ANSL
325   FORMAT(A1)
C
C
      IF(ANSL.EQ.'N') THEN
        GOTO 17
      ELSE
        CONTINUE
      ENDIF
      IF(ANSL.EQ.'Y') THEN
        GOTO 18
      ELSE
        GOTO 3336
18    ENDIF
      GOTO 1236
20    CALL CLS
126   WRITE(*,326)
326   FORMAT(//////,'  YOU SELECTED 4 COPPER LAYERS, GIVING YOU THE FOLL
     +OWING ALTERNATIVES:')
      WRITE(*,327)
327   FORMAT(//,'   1.  5 BY 18 PROVIDES A 1:3.6 RATIO WITH 720 NODES.
     +',/,'   2.  9 BY 10 PROVIDES A 1:1.11 RATIO WITH 720 NODES.',/,
     +'   3.  8 BY 10 PROVIDES A 1:1.25 RATIO WITH 640 NODES.',/,
     +'   4.  13 BY 6 PROVIDES A 1:1 RATIO WITH 624 NODES.',//,
     +'   PLEASE SELECT A NUMBER 1 THROUGH 4:     ',2X, )
      READ(*,328)LAY4
328   FORMAT(A1)
      IF (LAY4.NE.'1'.AND.LAY4.NE.'2'.AND.LAY4.NE.
     +'3'.AND.LAY4.NE.'4') THEN
      GOTO 20
      ELSE
3337  CALL CLS
      WRITE(*,329)LAY4
329   FORMAT(//////,'  YOU SELECTED NUMBER 'A1' OF THE FOLLOWING 4 ALTER
     +NATIVES:',///,
     +'   1.  5 BY 18 PROVIDES A 1:3.6 RATIO WITH 720 NODES.',/,
     +'   2.  9 BY 10 PROVIDES A 1:1.11 RATIO WITH 720 NODES.',/,
     +'   3.  8 BY 10 PROVIDES A 1:1.25 RATIO WITH 640 NODES.',/,
     +'   4.  13 BY 6 PROVIDES A 1:1 RATIO WITH 624 NODES.',///)
      ENDIF
21    WRITE(*,330)
330   FORMAT('  IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N FO
     +R NO:    ',2X, )
```

```
      READ(*,331)ANSL
331   FORMAT(A1)
C
C
      IF(ANSL.EQ.'N') THEN
        GOTO 20
      ELSE
        CONTINUE
      ENDIF
      IF(ANSL.EQ.'Y') THEN
        GOTO 22
      ELSE
        GOTO 3337
22    ENDIF
      GOTO 1237

1234  CONTINUE
      IF(LAY1.EQ.'1')THEN
           NWIDE=10
           NDEEP=36
           NPL=360
      ELSEIF(LAY1.EQ.'2')THEN
           NWIDE=10
           NDEEP=35
           NPL=350
      ELSEIF(LAY1.EQ.'3')THEN
           NWIDE=12
           NDEEP=25
           NPL=300
      ELSEIF(LAY1.EQ.'4')THEN
           NWIDE=15
           NDEEP=20
           NPL=300
      ENDIF
         GOTO 1241
1235  CONTINUE
      IF(LAY2.EQ.'1')THEN
           NWIDE=10
           NDEEP=18
           NPL=180
      ELSEIF(LAY2.EQ.'2')THEN
           NWIDE=9
           NDEEP=20
           NPL=180
      ELSEIF(LAY2.EQ.'3')THEN
           NWIDE=8
           NDEEP=20
           NPL=160
      ELSEIF(LAY2.EQ.'4')THEN
           NWIDE=10
           NDEEP=15
           NPL=150
      ENDIF
         GOTO 1241
1236  CONTINUE
      IF(LAY3.EQ.'1')THEN
```

```
             NWIDE=5
             NDEEP=24
             NPL=120
        ELSEIF(LAY3.EQ.'2')THEN
             NWIDE=10
             NDEEP=10
             NPL=100
        ELSEIF(LAY3.EQ.'3')THEN
             NWIDE=12
             NDEEP=10
             NPL=120
        ELSEIF(LAY3.EQ.'4')THEN
             NWIDE=8
             NDEEP=15
             NPL=120
        ENDIF
           GOTO 1241
1237    CONTINUE
        IF(LAY4.EQ.'1')THEN
             NWIDE=5
             NDEEP=18
             NPL=90
        ELSEIF(LAY4.EQ.'2')THEN
             NWIDE=9
             NDEEP=10
             NPL=90
        ELSEIF(LAY4.EQ.'3')THEN
             NWIDE=8
             NDEEP=10
             NPL=80
        ELSEIF(LAY4.EQ.'4')THEN
             NWIDE=13
             NDEEP=6
             NPL=78
        ENDIF
           GOTO 1241
C
C      EPOXY CHARACTERISTICS
C
1241    CALL CLS
        WRITE(*,8000)
8000    FORMAT(///,'*****************************************************
       +********************',/,
       +'****************************EPOXY CHARACTERISTICS****************
       +*********',/,
       +'****************************************************************
       +*********',/,)
C
C      PROVIDE CORRECT UNIT ABBREVIATIONS
C
        IF(ANSN.EQ.'S') THEN
          WRITE(*,5400)
5400      FORMAT('  ALL ENTRIES ARE IN SI NOTATION.',/)
          UL='cm'
          UK='Watts/cm/C'
          UT='C'
```

```
        UW='gm'
      ELSEIF(ANSN.EQ.'E') THEN
        WRITE(*,5401)
5401    FORMAT('  ALL ENTRIES ARE IN ENGLISH NOTATION',/)
        UL='in'
        UK='Btu/hr/F'
        UT='F'
        UW='oz'
      ENDIF
C
C
C
      WRITE(*,5402) UL
5402  FORMAT(/,'  ENTER EPOXY LAYER LENGTH (',A2,'): ',2X, )
      READ *,EL
C
      WRITE(*,5403) UL
5403  FORMAT(/,'  ENTER EPOXY LAYER WIDTH (',A2,'):  ',2X, )
      READ *,EW
4337  WRITE(*,6403)
6403  FORMAT(//,'  THE EPOXY LAYER THICKNESS WILL DEFAULT TO 0.0625 in (
     +0.15875 cm)',/,'  DO YOU WANT TO CHANGE THE EPOXY THICKNESS? (Y OR
     + N): ',2X, )
      READ(*,5799) ANSQ
5799  FORMAT(A1)
      IF(ANSQ.EQ.'Y') THEN
      GOTO 5798
      ELSEIF(ANSQ.EQ.'N'.AND.ANSN.EQ.'S') THEN
      DELE=0.15875
      GOTO 4338
      ELSEIF(ANSQ.EQ.'N'.AND.ANSN.EQ.'E') THEN
      DELE=0.0625
      GOTO 4338
      ELSE
      GOTO 4337
      ENDIF
5798  WRITE(*,5404) UL
5404  FORMAT(/,'  ENTER EPOXY LAYER THICKNESS (',A2,'):  ',2X, )
      READ *,DELE
C
4338  WRITE(*,5405) UK
5405  FORMAT(/,'  ENTER EPOXY LAYER THERMAL CONDUCTIVITY (',A10,'):  ',2
     +X, )
      READ *,UKE
5459  CALL CLS
C
C    MAKE CHANGES OR CORRECTIONS TO EPOXY ENTRIES
C
      WRITE(*,5406)
5406  FORMAT(///,'      YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE EPOXY
     + LAYER:',/,)
5407  WRITE(*,89343) EL,UL
89343 FORMAT(/,'         1.) LENGTH:     ',F9.4,1X,A2, )
      WRITE(*,89344) EW,UL
89344 FORMAT(/,'         2.) WIDTH:      ',F9.4,1X,A2, )
```

```
      WRITE(*,89345) DELE,UL
89345 FORMAT(/,'        3.) THICKNESS:  ',F9.4,1X,A2, )
      WRITE(*,89346) UKE,UK
89346 FORMAT(/,'        4.) k:          ',F9.4,1X,A11,///, )
C
      WRITE(*,5408)
5408  FORMAT(/,'      DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR YES
     + AND N FOR NO:    ',2X, )
      READ(*,5409)ANSE
5409  FORMAT(A1)
C
C
      IF(ANSE.EQ.'Y') THEN
5441     CALL CLS
         WRITE(*,5410) EL,UL
5410     FORMAT(////,'    THE CURRENT ENTRY FOR LENGTH IS: ',F9.4,1X,A2)
         WRITE(*,5411)
5411     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE LENGTH? (Y OR N):
     +',2X, )
         READ(*,5412)ANSEL
5412     FORMAT(A1)
         PRINT *
            IF(ANSEL.EQ.'Y') THEN
            WRITE(*,5422) UL
5422  FORMAT(/,'      ENTER THE EPOXY LENGTH (',A2,'):   ',2X, )
            READ *,EL
            ELSEIF(ANSEL.EQ.'N') THEN
            GOTO 5440
            ELSE
            GOTO 5441
            ENDIF
C
5440        CALL CLS
            WRITE(*,5442) EW,UL
5442  FORMAT(////,'    THE CURRENT ENTRY FOR WIDTH IS: ',F9.4,1X,A2)
         WRITE(*,5443)
5443     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE WIDTH?  (Y OR N):
     +',2X, )
         READ(*,5444)ANSEW
5444     FORMAT(A1)
         PRINT *
            IF(ANSEW.EQ.'Y') THEN
            WRITE(*,5445) UL
5445  FORMAT(/,'      ENTER THE EPOXY WIDTH (',A2,'):   ',2X, )
            READ *,EW
            ELSEIF(ANSEW.EQ.'N') THEN
            GOTO 5446
            ELSE
            GOTO 5440
            ENDIF
C
5446        CALL CLS
            WRITE(*,5447) DELE,UL
5447  FORMAT(////,'    THE CURRENT ENTRY FOR THICKNESS IS: ',F9.4,1X,A2)
         WRITE(*,5448)
5448     FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THICKNESS?  (Y OR N
```

```
     +):  ',2X, )
         READ(*,5449)ANSET
5449     FORMAT(A1)
         PRINT *
            IF(ANSET.EQ.'Y') THEN
            WRITE(*,5450) UL
5450  FORMAT(/,'     ENTER THE EPOXY THICKNESS (',A2,'):  ',2X, )
            READ *,DELE
            ELSEIF(ANSET.EQ.'N') THEN
            GOTO 5451
            ELSE
            GOTO 5446
            ENDIF
C
5451        CALL CLS
            WRITE(*,5452) UKE,UK
5452  FORMAT(////,'     THE CURRENT ENTRY FOR THERMAL CONDUCTIVITY IS: ',
     +F9.4,1X,A10)
         WRITE(*,5453)
5453     FORMAT(/,'      WOULD YOU LIKE TO CHANGE THE THERMAL CONDUCTIVIT
     +Y?  (Y OR N):   ',2X, )
         READ(*,5454)ANSEK
5454     FORMAT(A1)
         PRINT *
            IF(ANSEK.EQ.'Y') THEN
            WRITE(*,5455) UK
5455  FORMAT(/,'     ENTER THE THERMAL CONDUCTIVITY (',A10,').  ',2X, )
            READ *,UKE
            ELSEIF(ANSEK.EQ.'N') THEN
            GOTO 5456
            ELSE
            GOTO 5451
            ENDIF
5456        CALL CLS
            WRITE(*,5457)
5457  FORMAT(///,'        YOU HAVE MADE THE FOLLOWING CORRECTIONS TO THE
     + EPOXY ENTRIES:',//)
      GOTO 5407
C
C
      ELSEIF(ANSE.EQ.'N') THEN
      GOTO 5458
      ELSE
      GOTO 5459
      ENDIF
C
C
C
C    CALL SUBROUTINE COPPER
C
C
C
5458  CALL COPPER(EW,EL,ANSN,UK,SELECT,T1L1,T2L1,T2L2,T3L1,T3L2,T3L3,T4L
     +1,T4L2,T4L3,T4L4,SPEVAL,UKC,CL,CW)
C
C
```

```
C
C    CALL SUBROUTINE PCBS1
C
C
C
31180 CALL PCBS1
C
C
C    CALL SUBROUTINE PCBS2
C
C
C
      CALL PCBS2(THEAT,THPN,AHEAT,NHEAT,NPL,NWIDE,NDEEP,HEAT,IH,JH,ANSN,
     +EL,EW)
C
C
C    CALL SUBROUTINE PCBS3
C
C
      CALL PCBS3(EW,EL,CW,CL,UKE,DELE,UKC,SELECT,T1L1,T2L1,T2L2,T3L1,T3L
     +2,T3L3,T4L1,T4L2,T4L3,T4L4,NWIDE,NDEEP,NPL,IH,JH,HEAT,COEF,IBT,UPR
     +T,LWRT,RT,LT,FT,BT,ANSN)
      END
C***********************************************************************
C     SUBROUTINE INTRO
C
C
C    TITLE:      MODEL BUILDER
C    SUBROUT:    INTRO
C    AUTHOR:     LT STEVE GLASER
C    DATE:       09 JUL 1991
C    COMPILER:   MICROSOFT VERSION 4.01
C    LINKER:     MICROSOFT VERSION 3.55
C
C
C    THIS SUBROUTINE PROVIDES THE USER WITH AN OVERVIEW OF THE THERMAL
C    ANALYZER MODEL BUILDER.
C
C
C

      COMMON IBT,UPRT,LWRT,RT,LT,FT,BT,UT,ANSWER
C
C
C
C
C     DEFINE ONE VARIABLE CHARACTER VALUES
C
      CHARACTER*1 ANSWER

C
C    PROVIDE THE USER WITH A PROGRAM OVERVIEW
C
C
C
C
```

```
C
C
7002  FORMAT (A1)
      CALL CLS
      WRITE(*,7003)
7003  FORMAT(//////,'                 ********* OVERVIEW **********     ',//
     +'  THIS PROGRAM PERFORMS A NODAL ANALYSIS OF A PRINTED    ',/,
     +'  CIRCUIT BOARD CONTAINING UP TO FOUR COPPER LAYERS      ',/,
     +'  (WITH EPOXY LAYERS IN BETWEEN).  THE OUTPUT CONSISTS   ',/,
     +'  OF UP TO 720 COEFFICIENTS THAT CONTRIBUTE TO THE       ',/,
     +'  DETERMINATION OF THE TEMPERATURE DISTRIBUTION          ',/,
     +'  OF THE PRINTED CIRCUIT BOARD (PCB) WHEN FED INTO THE    ',/,
     +'  THERMAL ANALYZER.  ',//,
     +'  THE FOLLOWING IS AN OUTLINE OF THE MAJOR      ',/,
     +'  SECTIONS OF THIS PROGRAM AND WHAT ENTRIES ARE          ',/,
     +'  REQUIRED OF THE USER.                                  ',/,
     +'                                                         ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS     ',/,
     +'                                                         ',/,
     +'               PRESS <ENTER> TO CONTINUE                 ',2X, )
      READ(*,7002)ANSWER
C
      CALL CLS
C
      WRITE(*,7004)
7004  FORMAT(//////,'                 ********* OVERVIEW **********     ',//
     +'  A. DATA OUTPUT FILE:                                   ',/,
     +'                                                         ',/,
     +'  THIS PROGRAM GENERATES AN OUTPUT DATA FILE  WHICH IS   ',/,
     +'  TO BE THE INPUT TO THE THERMAL ANALYZER.               ',/,
     +'  DURING THE COURSE OF THE PROGRAM, THE USER WILL BE     ',/,
     +'  ASKED TO PROVIDE A NAME FOR THE OUTPUT FILE.           ',/,
     +'                                                         ',/,
     +'  WHEN PROMPTED PLEASE ENTER THE NAME OF THE OUTPUT      ',/,
     +'  DATA FILE.  THE DATA FILE NAME SHOULD BE NO LONGER     ',/,
     +'  THAN SIX LETTERS, AND MAY NOT HAVE ANY SPACES.         ',/,
     +'                                                         ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS     ',/,
     +'                                                         ',/,
     +'                PRESS <ENTER> TO CONTINUE                 ',2X, )
      READ(*,7002)ANSWER
C
      CALL CLS
      WRITE(*,7005)
7005  FORMAT(//////,'                 ********* OVERVIEW **********     ',//
     +'  B. STRUCTURE PHYSICAL CHARACTERISTICS                  ',/,
     +'                                                         ',/,
     +'  1.   YOU WILL BE ASKED TO SELECT UNIT TYPE             ',/,
     +'       (SI OR ENGLISH).                                  ',/,
     +'  2.   THE PRINTED CIRCUIT BOARD IS MAINLY COMPOSED OF   ',/,
     +'       ALTERNATING COPPER AND EPOXY LAYERS.  THE PROGRAM',/,
     +'       IS DESIGNED TO PROVIDE THE USER WITH A RANGE      ',/,
     +'       OF 1 TO 4 COPPER LAYERS. IT IS ASSUMED THAT EACH ',/,
     +'       COPPER LAYER LIES BETWEEN EPOXY LAYERS.           ',/,
     +'       DEPENDING ON THE NUMBER OF COPPER LAYERS SELECTED,',/,
     +'       THE PROGRAM WILL ALLOW THE USER TO CHOOSE FROM    ',/,
```

```
     +'      FOUR ALTERNATIVE NODAL ASPECT RATIOS.                  ',/,
     +'                                                             ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS         ',/,
     +'                                                             ',/,
     +'                PRESS <ENTER> TO CONTINUE                    ',2X, )
      READ(*,7002)ANSWER
      CALL CLS
      WRITE(*,7006)
7006  FORMAT(//////,'              ********* OVERVIEW **********     ',//
     +'  B. STRUCTURE PHYSICAL CHARACTERISTICS (CONTINUED)          ',/,
     +'                                                             ',/,
     +'  3.   AFTER SELECTING THE DESIRED NUMBER OF COPPER          ',/,
     +'       LAYERS AND NODAL ASPECT RATIO, THE PROGRAM WILL       ',/,
     +'       THEN ASK THE USER TO PROVIDE LAYER CHARACTERISTICS.   ',/,
     +'       EPOXY AND COPPER  LENGTH, THICKNESS, AND WIDTH,       ',/,
     +'       AS WELL AS CONSTANTS CONSTITUTE THE QUERIES.          ',/,
     +'       THE PROGRAM WILL ALSO ASK THE USER TO PROVIDE         ',/,
     +'       THE PERCENT COVERAGE FOR EACH COPPER LAYER.           ',/,
     +'                                                             ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS         ',/,
     +'                                                             ',/,
     +'                PRESS <ENTER> TO CONTINUE                    ',2X, )
      READ(*,7002)ANSWER
      CALL CLS
      WRITE(*,7007)
7007  FORMAT(//////,'              ********* OVERVIEW **********     ',//
     +'  C. INTITIAL AND AMBIENT TEMPERATURES                       ',/,
     +'                                                             ',/,
     +'  1.   AFTER SELECTING THE DESIRED NUMBER OF COPPER          ',/,
     +'       LAYERS AND NODAL ASPECT RATIO, THE PROGRAM WILL       ',/,
     +'       THEN ASK THE USER TO PROVIDE TEMPERATURES FOR THE .   ',/,
     +'       BOARD.                                                ',/,
     +'                                                             ',/,
     +'                                                             ',/,
     +'                                                             ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS         ',/,
     +'                                                             ',/,
     +'                PRESS <ENTER> TO CONTINUE                    ',2X, )
      READ(*,7002)ANSWER
      CALL CLS
      WRITE(*,7008)
7008  FORMAT(//////,'              ********* OVERVIEW **********     ',//
     +'  D. HEAT INPUT.                                             ',/,
     +'                                                             ',/,
     +'  1.   HEAT INJECTION OCCURS ONLY ON THE UPPER COPPER        ',/,
     +'       LAYER.  THIS PROGRAM SUPPLIES THE USER FOR            ',/,
     +'       ALTERNATIVE METHODS FOR ENTERING HEAT.                ',/,
     +'                                                             ',/,
     +'         A.) TOTAL HEAT OVER SURFACE.                   ',/,
     +'         B.) AVERAGE HEAT PER UNIT AREA                 ',/,
     +'         C.) INPUT HEAT NODE BY NODE                    ',/,
     +'         D.) NO HEAT INPUT                              ',/,
     +'                                                        ',/,
     +'  PLEASE NOTE: ENTRIES MUST BE IN UPPER CASE LETTERS         ',/,
     +'                                                             ',/,
     +'     ****THIS CONCLUDES THE PROGRAM OVERVIEW****             ',//,
```

```
     +'               PRESS <ENTER> TO CONTINUE                        ',2X, )
      READ(*,7002)ANSWER
      CALL CLS
      END
      SUBROUTINE PCBS1
C ***********************************************************************
C
C    TITLE:      MODEL BUILDER
C    SUBROUT:    PCBS1
C    AUTHOR:     LT STEVE GLASER
C    DATE:       09 MAY 1991
C    COMPILER:   MICROSOFT VERSION 4.01
C    LINKER:     MICROSOFT VERSION 3.55
C
C    THIS SUBROUTINE HANDLES THE TEMPERATURE INPUTS TO THE PRINTED CIRCUIT
C    BOARD.
C
C
C
C
      COMMON IBT,UPRT,LWRT,RT,LT,FT,BT,UT,ANSWER
C
C    DEFINE REAL VARIABLES
C
C
      REAL IBT,UPRT,LWRT,RT,LT,FT,BT
C
C    DEFINE ONE CHARACTER VARIABLES
C
      CHARACTER*1 ANSTEM,ANSTEB,ANSTEU,ANSTEL,ANSTER,ANSLFT,ANSFRT,ANSBC
     +K,UT
C
31180 CALL CLS
      WRITE(*,60000)
60000 FORMAT(///,'**********************************************************
     +*********************',/,
     +'********************AMBIENT TEMPERATURE INPUTS*********************
     +*********',/,
     +'*****************************************************************
     +*********',//,)
C
C
      WRITE(*,60001) UT
60001 FORMAT(/,'    ENTER THE INITIAL BOARD TEMPERATURE (',A1,'): ',2X,
     +)
      READ *,IBT
C
C
      WRITE(*,60002) UT
60002 FORMAT(/,'    ENTER THE UPPER SURFACE AMBIENT TEMPERATURE (',A1,')
     +: ',2X, )
      READ *,UPRT
C
C
C
```

```
      WRITE(*,60003) UT
60003 FORMAT(/,'    ENTER THE LOWER SURFACE AMBIENT TEMPERATURE (',A1,')
     +: ',2X, )
      READ *,LWRT
C
C
      WRITE(*,60004) UT
60004 FORMAT(/,'    ENTER THE RIGHT SURFACE AMBIENT TEMPERATURE (',A1,')
     +: ',2X, )
      READ *,RT
C
C
C
      WRITE(*,60005) UT
60005 FORMAT(/,'    ENTER THE LEFT SURFACE AMBIENT TEMPERATURE (',A1,'):
     + ',2X, )
      READ *,LT
C
C
      WRITE(*,60006) UT
60006 FORMAT(/,'    ENTER THE FRONT SURFACE AMBIENT TEMPERATURE (',A1,')
     +: ',2X, )
      READ *,FT
C
C
C
      WRITE(*,60007) UT
60007 FORMAT(/,'    ENTER THE REAR SURFACE AMBIENT TEMPERATURE (',A1,'):
     + ',2X, )
      READ *,BT
C
C
C    REVIEW THE TEMPERATURE ENTRIES
C
60052 CALL CLS
      WRITE(*,60008) IBT,UT
60008 FORMAT(////,'    YOU HAVE MADE THE FOLLOWING AMBIENT TEMPERATURE E
     +NTRIES: ',////,'    1. INITIAL BOARD TEMPERATURE: ',1X,F9.3,1X,A1)
      WRITE(*,60009) UPRT,UT
60009 FORMAT('    2. UPPER AMBIENT TEMPERATURE: ',1X,F9.3,1X,A1)
      WRITE(*,60010) LWRT,UT
60010 FORMAT('    3. LOWER AMBIENT TEMPERATURE: ',1X,F9.3,1X,A1)
      WRITE(*,60011) RT,UT
60011 FORMAT('    4. RIGHT AMBIENT TEMPERATURE: ',1X,F9.3,1X,A1)
      WRITE(*,60012) LT,UT
60012 FORMAT('    5. LEFT AMBIENT TEMPERATURE:  ',1X,F9.3,1X,A1)
      WRITE(*,60013) FT,UT
60013 FORMAT('    6. FRONT AMBIENT TEMPERATURE: ',1X,F9.3,1X,A1)
      WRITE(*,60014) BT,UT
60014 FORMAT('    7. REAR AMBIENT TEMPERATURE:  ',1X,F9.3,1X,A1,//)
C
C
C
      WRITE(*,60015)
60015 FORMAT(/,'    DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR YES A
     +ND N FOR NO: ',2X, )
```

```
      READ(*,60016)ANSTEM
60016 FORMAT(A1)
C
C
C    MAKE CORRECTIONS BOARD TEMPERATURES
C
C
      IF(ANSTEM.EQ.'Y') THEN
60022   CALL CLS
        WRITE(*,60017) IBT,UT
60017   FORMAT(///,'    THE INITIAL BOARD TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60018)
60018   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60019) ANSTEB
60019   FORMAT(A1)
           IF(ANSTEB.EQ.'Y') THEN
             WRITE(*,60020) UT
60020        FORMAT(/,'    ENTER THE NEW VALUE (',A1,'): ',2X, )
             READ *,IBT
           ELSEIF(ANSTEB.EQ.'N') THEN
             GOTO 60021
             ELSE
             GOTO 60022
           ENDIF
C
60021    CALL CLS
         WRITE(*,70022) UPRT,UT
70022   FORMAT(///,'    THE UPPER AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60023)
60023   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60024) ANSTEU
60024   FORMAT(A1)
           IF(ANSTEU.EQ.'Y') THEN
             WRITE(*,60025) UT
60025        FORMAT(/,'    ENTER THE NEW VALUE (',A1,'): ',2X, )
             READ *,UPRT
           ELSEIF(ANSTEU.EQ.'N') THEN
             GOTO 60026
             ELSE
             GOTO 60021
           ENDIF
C
C
60026    CALL CLS
         WRITE(*,60027) LWRT,UT
60027   FORMAT(///,'    THE LOWER AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60028)
60028   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60029) ANSTEL
60029   FORMAT(A1)
           IF(ANSTEL.EQ.'Y') THEN
             WRITE(*,60030) UT
60030        FORMAT(/,'    ENTER THE NEW VALUE (',A1,'): ',2X, )
```

```
          READ *,LWRT
        ELSEIF(ANSTEL.EQ.'N') THEN
          GOTO 60031
          ELSE
          GOTO 60026
        ENDIF
C
C
C
60031   CALL CLS
        WRITE(*,60032) RT,UT
60032   FORMAT(///,'    THE RIGHT AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60033)
60033   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60034) ANSTER
60034   FORMAT(A1)
          IF(ANSTER.EQ.'Y') THEN
            WRITE(*,60035) UT
60035       FORMAT(/,'    ENTER THE NEW VALUE (',A1,'): ',2X, )
            READ *,RT
          ELSEIF(ANSTER.EQ.'N') THEN
            GOTO 60036
            ELSE
            GOTO 60031
          ENDIF
C
C
C
C
60036   CALL CLS
        WRITE(*,60037) LT,UT
60037   FORMAT(///,'    THE LEFT AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60038)
60038   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60039) ANSLFT
60039   FORMAT(A1)
          IF(ANSLFT.EQ.'Y') THEN
            WRITE(*,60040) UT
60040       FORMAT(/,'    ENTER THE NEW VALUE (',A1,'): ',2X, )
            READ *,LT
          ELSEIF(ANSLFT.EQ.'N') THEN
            GOTO 60041
            ELSE
            GOTO 60036
          ENDIF
C
C
C
C
60041   CALL CLS
        WRITE(*,60042) FT,UT
60042   FORMAT(///,'    THE FRONT AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60043)
60043   FORMAT(/,'    WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
```

```
     +X, )
        READ(*,60044) ANSFRT
60044   FORMAT(A1)
           IF(ANSFRT.EQ.'Y') THEN
             WRITE(*,60045) UT
60045        FORMAT(/,'     ENTER THE NEW VALUE (',A1,'): ',2X, )
             READ *,FT
           ELSEIF(ANSFRT.EQ.'N') THEN
             GOTO 60046
             ELSE
             GOTO 60041
           ENDIF
C
C
C
C
60046    CALL CLS
         WRITE(*,60047) BT,UT
60047   FORMAT(///,'     THE REAR AMBIENT TEMPERATURE IS: ',F9.3,1X,A1)
        WRITE(*,60048)
60048   FORMAT(/,'     WOULD YOU LIKE TO CHANGE THIS VALUE? (Y OR N): ',2
     +X, )
        READ(*,60049) ANSBCK
60049   FORMAT(A1)
           IF(ANSBCK.EQ.'Y') THEN
             WRITE(*,60050) UT
60050        FORMAT(/,'     ENTER THE NEW VALUE (',A1,'): ',2X, )
             READ *,BT
           ELSEIF(ANSBCK.EQ.'N') THEN
             CONTINUE
             ELSE
             GOTO 60046
           ENDIF
      ELSEIF(ANSTEM.EQ.'N') THEN
        GOTO 60051
      ELSE
        GOTO 60052
      ENDIF
60051 END
      SUBROUTINE PCBS2(THEAT,THPN,AHEAT,NHEAT,NPL,NWIDE,NDEEP,HEAT,IH,JH
     +,ANSN,EL,EW)
C
C
C     TITLE:      MODEL BUILDER
C     SUBROUT:    PCBS2 -- HEAT INPUT SUBROUTINE
C     AUTHOR:     LT STEVE GLASER
C     DATE:       09 MAY 1991
C     COMPILER:   MICROSOFT VERSION 4.01
C     LINKER:     MICROSOFT VERSION 3.55
C
C     THIS SUBROUTINE HANDLES THE HEAT INPUTS TO THE PRINTED CIRCUIT BOARD
C
C     DEFINE REAL VARIABLES
C
      REAL THEAT,THPN,AHEAT,NHEAT,SL,SW
C
```

```
C
C     DEFINE INTEGERS
C
       INTEGER NWIDE,NDEEP,NPL,NUM,NUMA,CH,H,TOTNOD,NC,NN,I,J,IM
C
C
C     DEFINE ONE CHARACTER VARIABLES
C
       CHARACTER*1 SELH,ANSH,ANSN,ATH,ANSHA,AHN,DUMMY
C
C     DEFINE CHARACTER VARIABLES
C
       CHARACTER UH*13,UAH*6
C
C     DEFINE REAL MATRICES
C
       REAL HEAT(100,100),COEFF(740,9)
C
C     DEFINE INTEGER VARIABLES
C
       INTEGER IH(1000),JH(1000),UMMY(1000)
C
C     PROVIDE A CORRELATION BETWEEN NODE NUMBERS AND MATRIX LOCATION
C
C
51     NUM=1
         DO 60 I=1,NPL/NWIDE
              DO 61 J=1,NWIDE
                  JH(NUM)=J
                  NUM=NUM+1
61            CONTINUE
60     CONTINUE
C
       NUMA=1
       CH=0
       H=1
       DO 62 I=1,NPL
          IH(NUMA)=H
          CH=CH+1
          IF(CH.EQ.NWIDE) THEN
            H=H+1
            CH=0
          ELSE
            CONTINUE
          ENDIF
       NUMA=NUMA+1
62     CONTINUE
C
C
C
1241   CALL CLS
       WRITE(*,8000)
8000   FORMAT(///,'*********************************************************
      +*********************',/,
      +'********************************HEAT INPUTS************************
      +*********',/,
```

```
     +'*****************************************************************
     +********',//,
     +'     HEAT INPUT TO THE PCB OCCURS ONLY ON THE UPPER COPPER',/,
     +'     LAYER.  HEAT  INPUT  IS ACCOMPLISHED  BY  ONE OF THE ',/,
     +'     FOLLOWING  METHODS: ',//,
     +'     1.  ENTER AS TOTAL HEAT APPLIED TO THE PCB',/,
     +'     2.  ENTER AS AVERAGE HEAT PER UNIT AREA    ',/,
     +'     3.  ENTER HEAT NODE BY NODE ',/,
     +'     4.  NO HEAT INPUT',//,
     +'     PLEASE SELECT NUMBER 1 THROUGH 4:   ',2X, )
C
      READ(*,5799) SELH
5799  FORMAT(A1)
      IF(SELH.EQ.'1'.OR.SELH.EQ.'2'.OR.SELH.EQ.'3'.OR.SELH.EQ.'4') THEN
444   WRITE(*,291) SELH
291   FORMAT(//,'     YOU HAVE SELECTED NUMER ',A1,' OF FOUR ALTERNATIVES
     +. ',/,'     IS THIS THE DESIRED SELECTION? (Y OR N): ',2X, )
      ELSE
      GOTO 1241
      ENDIF
      READ(*,5409)ANSH
5409  FORMAT(A1)
C
C
      IF(ANSH.EQ.'Y') THEN
5441    GOTO 63
      ELSEIF(ANSH.EQ.'N') THEN
        GOTO 51
      ELSE
        CALL CLS
        WRITE(*,8000)
        GOTO 444
63      ENDIF
C
C
C    DETERMINE UNIT FOR HEAT INPUT
C
      IF ((ANSN.EQ.'E').AND.(SELH.EQ.'2')) THEN
         UH = 'Btu/(hr*in                   2)'
         UAH = 'Btu/hr'
      ELSEIF ((ANSN.EQ.'E').AND.(SELH.EQ.'1'.OR.SELH.EQ.'3')) THEN
         UH = 'Btu/hr'
      ELSEIF ((ANSN.EQ.'S').AND.(SELH.EQ.'2')) THEN
         UH = 'Watts/(cm                    2)'
         UAH = 'Watts'
      ELSEIF ((ANSN.EQ.'S').AND.(SELH.EQ.'1'.OR.SELH.EQ.'3')) THEN
         UH = 'Watts'
      ENDIF
C
C
C    ALLOW FOR RE-SELECTION OF HEAT INPUT METHOD OR CONTINUE WITH
C    INITIAL SELECTION.
C
C    CHOICE #1
C
C
```

```
      CALL CLS
      IF (SELH.EQ.'1'.AND.ANSN.EQ.'E') THEN
989   WRITE(*,988) UH
988   FORMAT(///,'        YOU HAVE SELECTED TO INPUT HEAT AS A TOTAL HEAT
     +',/,'       APPLIED TO THE SURFACE. ',//,
     +'       ENTER TOTAL HEAT APPLIED TO THE SURFACE (',A6,'): ',2X, )
      READ *,THEAT
990   WRITE(*,991)
991   FORMAT(/,'       IS THIS THE CORRECT ENTRY? (Y OR N):   ',2X, )
      READ(*,992)ATH
992   FORMAT(A1)
C
C     MAKE ENTRY AND ALLOW FOR CORRECTIONS
C
      IF (ATH.EQ.'Y') THEN
         THPN=THEAT/NPL
         WRITE(*,993) THPN,UH
993      FORMAT(/,'       TOTAL HEAT PER NODE IS: ',F9.4,1X,A13)
         WRITE(*,9323)
9323     FORMAT(//,'                            <PRESS ENTER TO CONTINUE>')
         READ(*,9324)DUMMY
9324     FORMAT(A1)
C
C
C     FILL HEAT MATRIX WITH DESIRED VALUES
C
C
        DO 994 I=1,NDEEP
           DO 995 J=1,NWIDE
              HEAT(I,J)=THPN
995        CONTINUE
994     CONTINUE
C
C
        ELSEIF(ATH.EQ.'N') THEN
           CALL CLS
           GOTO 989
        ELSE
           CALL CLS
           WRITE(*,996) THEAT,UH
996        FORMAT(////,'       TOTAL HEAT APPLIED TO THE SURFACE IS: ',F9
     +.4,1X,A13, )
           GOTO 990
      ENDIF
C
C
C     CHOICE #2
C
C
      ELSEIF(SELH.EQ.'2') THEN
998        WRITE(*,997) UH
997        FORMAT(////,'       YOU HAVE SELECTED TO ENTER THE AVERAGE HEA
     +T OVER THE',/,'       UPPER PCB SURFACE.',//,'       ENTER THE DESIR
     +ED HEAT INPUT: (',A13,'): ',2X, )
           READ *,AHEAT
*C
```

```
C    MAKE ENTRY AND ALLOW FOR CORRECTION
C
1000  WRITE(*,999)
999   FORMAT(/,'      IS THIS THE CORRECT ENTRY? (Y OR N):   ',2X, )
      READ(*,1001)ANSHA
1001  FORMAT(A1)
      IF (ANSHA.EQ.'Y') THEN
         THPN=AHEAT*EL*EW/NPL
         WRITE(*,1002) THPN,UAH
1002     FORMAT(///,'      TOTAL HEAT PER NODE IS: ',F9.4,1X,A6)
         WRITE(*,4323)
4323     FORMAT(//,'                          <PRESS ENTER TO CONTINUE>')
         READ(*,4324)DUMMY
4324     FORMAT(A1)
C
C    FILL HEAT MATRIX WITH DESIRED VALUES
C
C
        DO 1003 I=1,NDEEP
           DO 1004 J=1,NWIDE
              HEAT(I,J)=THPN
1004       CONTINUE
1003    CONTINUE
C
C
        ELSEIF(ANSHA.EQ.'N') THEN
           CALL CLS
           GOTO 998
        ELSE
           CALL CLS
           WRITE(*,1006) AHEAT,UH
1006       FORMAT(////,'      AVERAGE HEAT OVER PCB SURFACE IS: ',F9.4,1
     +X,A13, )
           GOTO 1000
      ENDIF
C
C    CHOICE #3
C
C
      ELSEIF(SELH.EQ.'3') THEN
1200     WRITE(*,1201)
1201     FORMAT(////,'      YOU HAVE SELECTED TO ENTER THE HEAT NODALLY'
     +,//,'      ENTER THE TOTAL NUMBER OF NODES DESIGNATED FOR HEAT INP
     +UT: ',2X, )
         READ *,TOTNOD
C
C
C    THIS IS DONE NODE BY NODE. GET NUMBER OF ENTRIES AND THEN LOOP UNTIL
C    ALL ENTRIES HAVE BEEN MADE.
C
C
C    TELL USER MAXIMUM NUMBER OF ENTRIES POSSIBLE
C
      IF (TOTNOD.GT.NPL) THEN
      WRITE(*,1202) NPL
1202  FORMAT(///,'      THE MAXIMUM ENTRY IS: ',I4)
```

```
      WRITE(*,2239)
2239  FORMAT(///,'       PLEASE PRESS <ENTER> TO CONTINUE    ')
      READ(*,2240) DUMMY
2240  FORMAT(A1)
      CALL CLS
      GOTO 1200
      ENDIF
C
C    MAKE ENTRIES
C
C
      DO 1203 I=1,TOTNOD
             NC=I
1204         CALL CLS
             WRITE(*,1205) NC,TOTNOD
1205         FORMAT(///,'      THIS IS NUMBER ',I3,' OF ',I3,' ENTRIES')
             WRITE(*,5345)
5345         FORMAT(//,'      ENTER THE NODE NUMBER FOR HEAT INPUT:  ',2
     +X, )
             READ (*,8032) NN
8032         FORMAT(I4)
             UMMY(I)=NN
             IF (NN.EQ.0.OR.NN.GT.NPL) THEN
               GOTO 1204
             ENDIF
             WRITE(*,1206) UH
1206         FORMAT(/,'      ENTER THE HEAT INPUT (',A6,'): ',2X, )
             READ *,NHEAT
             HEAT(IH(NN),JH(NN)) = NHEAT
1203  CONTINUE
C
C    PROVIDE OPPORTUNITY TO MAKE CORRECTIONS
C
C
1301  CALL CLS
      WRITE(*,1302) TOTNOD
1302  FORMAT(///,'       YOU HAVE MADE ',I3,' NODAL ENTRIES: ',/)
      DO 11203 IM=1,TOTNOD
      WRITE(*,11204) UMMY(IM),HEAT(IH(UMMY(IM)),JH(UMMY(IM))),UH
11204 FORMAT(1X,'      NODE NUMBER ',I4,':',1X,F9.4,1X,A6)
11203 CONTINUE
      WRITE(*,1303)
1303  FORMAT(/,'      DO YOU WISH TO MAKE ANY MORE ENTRIES OR CORRECTION
     +S? (Y OR N): ',2X, )
      READ(*,1304) AHN
1304  FORMAT(A1)
      IF(AHN.EQ.'Y') THEN
        CALL CLS
        GOTO 1200
      ELSEIF(AHN.EQ.'N') THEN
        GOTO 1305
      ELSE
        GOTO 1301
1305  ENDIF
      ELSEIF(SELH.EQ.'4') THEN
```

```
      GOTO 1306
1306  ENDIF
      END
      SUBROUTINE PCBS3(XEW,XEL,XCW,XCL,XUKE,XDELE,XUKC,SELECT,X1L1,X2L1,
     +X2L2,X3L1,X3L2,X3L3,X4L1,X4L2,X4L3,X4L4,XWIDE,XDEEP,XPL,XIH,XJH,X
     +EAT,XOEF,XIBT,XPRT,XWRT,XRT,XLT,XFT,XBT,XNITS)

C***********************************************************************
C
C     TITLE:       MODEL BUILDER
C     SUBROUTINE:  PCBS3
C     AUTHOR:      LT STEVE GLASER
C     DATE:        09 JUL 1991
C     COMPILER:    MICROSOFT VERSION 4.01
C     LINKER:      MICROSOFT VERSION 3.55
C
C           THIS SUBROUTINE IS THE "HEART" OF THE MODEL BUILDER.  THE
C           NODE COEFFICIENTS ARE CALCULATED IN THIS SUBROUTINE AND
C           PLACED IN MATRICES.  PCBS3 IN TURN ALSO CALLS OTHER
C           SUBROUTINES: S1, S2, S3, AND S4.  THESE SUBROUTINES CALLED
C           BY PCBS3 GENERATE THE OUTPUT DATA FILE FOR THE THERMAL
C           ANALYZER.
C
C
C
C     DEFINE REAL VARIABLES
C

      REAL XEL,XEW,XDELE,XUKE,XCL,XCW,X1L1,X2L1,X2L2,X3L1,X3L2,X3L3,X4L1
     +,X4L2,X4L3,X4L4,XUKC,EZB

      REAL EYLR,EFXB,EYY,EZC11,EZC21,EZC22,EZC31,EZC32,EZC33,EZC41,EZC42
     +,EZC43,EZC44,EXX,DELX,DELY

      REAL CYLR11,CYLR21,CYLR22,CYLR31,CYLR32,CYLR33,CYLR41,CYLR42,CYLR4
     +3,CYLR44
      REAL CFXB11,CFXB21,CFXB22,CFXB31,CFXB32,CFXB33,CFXB41,CFXB42,CFXB4
     +3,CFXB44
      REAL CYY11,CYY21,CYY22,CYY31,CYY32,CYY33,CYY41,CYY42,CYY43,CYY44

      REAL CXX11,CXX21,CXX22,CXX31,CXX32,CXX33,CXX41,CXX42,CXX43,CXX44

      REAL CZE11,CZE21,CZE22,CZE31,CZE32,CZE33,CZE41,CZE42,CZE43,CZE44

      REAL PC11,PC21,PC22,PC31,PC32,PC33,PC41,PC42,PC43,PC44

      REAL XEAT(100,100),XOEF(1000,12)

C
C     DEFINE CHARACTER VARIABLES
C
C
      CHARACTER*1 SELECT,ANS,XNITS
C
C     DEFINE INTEGER VARIABLES
```

```
C
C
      INTEGER XWIDE,XDEEP,XPL,NUMC,I,N,IB,USEL

      INTEGER XIH(1000),XJH(1000)
C
C***************************************************************************
C**********************DETERMINE COEFFICIENTS******************************
C***************************************************************************
C
C
C    DETERMINE INCREMENTAL MEASUREMENTS IN THE X AND Y DIRECTIONS
C
C
C    XNITS IS A CODE USED IN THE OUTPUT DATA FILE. 1 MEANS ENGLISH UNITS,
C    AND 2 MEANS SI UNITS
C
C
      IF (XNITS.EQ.'E') THEN
         USEL=1
      ELSE
         USEL=2
      ENDIF
C
C    DETERMINE INCREMENTS IN THE ARRAY
C
      DELX = XEL/XDEEP
      DELY = XEW/XWIDE
1111  CONTINUE
C
C
C    COPPER COVERAGE, ONE COPPER LAYER CASE
C
C
      IF(SELECT.EQ.'1') THEN
931   CALL CLS
C
C    ENTER PERCENT COPPER COVERAGE
C
C
      WRITE(*,3941)
3941  FORMAT(////,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR
     + LAYER 1: ',2X, )
      READ *,PC11
      IF (PC11.GT.100) THEN
      GOTO 931
      ENDIF
      PC11=PC11/100
C
C
C    COPPER COVERAGE, TWO COPPER LAYERS CASE
C
C
C
C
```

```
      ELSEIF(SELECT.EQ.'2') THEN
932   CALL CLS
      WRITE(*,3942)
3942  FORMAT(////,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR
     + LAYER 1: ',2X, )
      READ *,PC21
      IF (PC21.GT.100) THEN
      GOTO 932
      ENDIF
      PC21=PC21/100
933   WRITE(*,3943)
3943  FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 2: ',2X, )
      READ *,PC22
      IF (PC22.GT.100) THEN
      GOTO 933
      ENDIF
      PC22=PC22/100
C
C
C     COPPER COVERAGE, THREE COPPER LAYERS CASE
C
C
      ELSEIF(SELECT.EQ.'3') THEN
942   CALL CLS
      WRITE(*,4942)
4942  FORMAT(////,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR
     + LAYER 1: ',2X, )
      READ *,PC31
      IF (PC31.GT.100) THEN
      GOTO 942
      ENDIF
      PC31=PC31/100
433   WRITE(*,4943)
4943  FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 2: ',2X, )
      READ *,PC32
      IF (PC32.GT.100) THEN
      GOTO 433
      ENDIF
      PC32=PC32/100
633   WRITE(*,8943)
8943  FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 3: ',2X, )
      READ *,PC33
      IF (PC33.GT.100) THEN
      GOTO 633
      ENDIF
      PC33=PC33/100
C
C
C     COPPER COVERAGE, FOUR COPPER LAYERS CASE
C
C
      ELSEIF(SELECT.EQ.'4') THEN
242   CALL CLS
```

```
      WRITE(*,2942)
2942  FORMAT(////,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR
     + LAYER 1: ',2X, )
      READ *,PC41
      IF (PC41.GT.100) THEN
      GOTO 242
      ENDIF
      PC41=PC41/100
233   WRITE(*,2943)
2943  FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 2: ',2X, )
      READ *,PC42
      IF (PC42.GT.100) THEN
      GOTO 233
      ENDIF
      PC42=PC42/100
133   WRITE(*,1943)
1943  FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 3: ',2X, )
      READ *,PC43
      IF (PC43.GT.100) THEN
      GOTO 133
      ENDIF
      PC43=PC43/100
33    WRITE(*,43)
43    FORMAT(/,'       ENTER THE PERCENT COPPER COVERAGE (i.e. 67) FOR LA
     +YER 4: ',2X, )
      READ *,PC44
      IF (PC44.GT.100) THEN
      GOTO 33
      ENDIF
      PC44=PC44/100
      ENDIF
      CALL CLS
C
C
C
C     VERIFY THAT THE SELECTED ENTRIES ARE CORRECT
C
C
C

      IF(SELECT.EQ.'1') THEN
305   WRITE(*,300) PC11*100
300   FORMAT(/////'       YOU HAVE SELECTED ',F6.2,'% COVERAGE.',//,'
     +  IS THIS YOUR DESIRED ENTRY?  (Y OR N): ',1X, )
      READ (*,301)ANS
301   FORMAT(A1)
      IF(ANS.EQ.'Y') THEN
      GOTO 333
      ELSEIF(ANS.EQ.'N') THEN
      GOTO 1111
      ELSE
      CALL CLS
      GOTO 305
      ENDIF
```

```
      ELSEIF(SELECT.EQ.'2') THEN
306   WRITE(*,307) PC21*100,PC22*100
307   FORMAT(/////'       YOU HAVE SELECTED: ',F6.2,'% COVERAGE, LAYER 1'
     +,/,25X,F6.2,'% COVERAGE, LAYER 2.',//,'       IS THIS YOUR DESIRED
     + ENTRY?   (Y OR N): ',1X, )
      READ (*,308)ANS
308   FORMAT(A1)
      IF(ANS.EQ.'Y') THEN
      GOTO 333
      ELSEIF(ANS.EQ.'N') THEN
      GOTO 1111
      ELSE
      CALL CLS
      GOTO 306
      ENDIF
      ELSEIF(SELECT.EQ.'3') THEN
310   WRITE(*,311) PC31*100,PC32*100,PC33*100
311   FORMAT(/////'       YOU HAVE SELECTED: ',F6.2,'% COVERAGE, LAYER 1'
     +,/,25X,F6.2,'% COVERAGE, LAYER 2',/,25X,F6.2,'% COVERAGE, LAYER 3.
     +',//,'       IS THIS YOUR DESIRED ENTRY?   (Y OR N): ',1X, )
      READ (*,312)ANS
312   FORMAT(A1)
      IF(ANS.EQ.'Y') THEN
      GOTO 333
      ELSEIF(ANS.EQ.'N') THEN
      GOTO 1111
      ELSE
      CALL CLS
      GOTO 310
      ENDIF
      ELSEIF(SELECT.EQ.'4') THEN
320   WRITE(*,321) PC41*100,PC42*100,PC43*100,PC44*100
321   FORMAT(/////'       YOU HAVE SELECTED: ',F6.2,'% COVERAGE, LAYER 1'
     +,/,25X,F6.2,'% COVERAGE, LAYER 2',/,25X,F6.2,'% COVERAGE, LAYER 3'
     +,/,25X,F6.2,'% COVERAGE, LAYER 4.',//,'       IS THIS YOUR DESIRED
     + ENTRY?  (Y OR N): ',1X, )
      READ (*,322)ANS
322   FORMAT(A1)
      IF(ANS.EQ.'Y') THEN
      GOTO 333
      ELSEIF(ANS.EQ.'N') THEN
      GOTO 1111
      ELSE
      CALL CLS
      GOTO 320
      ENDIF
      ENDIF
333   CONTINUE
C
C
C
C     GENERATE CONSTANTS FOR THE EPOXY LAYERS
C
C     LEFT OF RIGHT EDGE TO OUTSIDE
C
C
```

```
      EYLR = 2*XUKE*DELX*XDELE/DELY
C
C
C    FRONT OR BACK TO OUTSIDE
C
C
      EFXB = 2*XUKE*DELY*XDELE/DELX
C
C
C
C    INNER MATRIX MOVEMENT IN THE Y DIRECTION
C
C
      EYY =  XUKE*DELX*XDELE/DELY
C
C
C
C
C    INNER MATRIX MOVEMENT IN THE X DIRECTION
C
C
C
      EXX =  XUKE*XDELE*DELY/DELX
C
C
C
C
C    EPOXY TO COPPER
C
C
C    ONE COPPER LAYER CASE
C
C
C
CC
      IF(SELECT.EQ.'1') THEN
      EZC11= 2*DELX*DELY/((XDELE/XUKE)+(X1L1*PC11/XUKC))
C
C
C    TWO COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      EZC21 = 2*DELX*DELY/((XDELE/XUKE)+(X2L1*PC21/XUKC))
      EZC22 = 2*DELX*DELY/((XDELE/XUKE)+(X2L2*PC22/XUKC))
C
C
C
C    THREE COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
```

```
      EZC31 = 2*DELX*DELY/((XDELE/XUKE)+(X3L1*PC31/XUKC))
      EZC32 = 2*DELX*DELY/((XDELE/XUKE)+(X3L2*PC32/XUKC))
      EZC33 = 2*DELX*DELY/((XDELE/XUKE)+(X3L3*PC33/XUKC))
C
C
C     FOUR COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      EZC41 = 2*DELX*DELY/((XDELE/XUKE)+(X4L1*PC41/XUKC))
      EZC42 = 2*DELX*DELY/((XDELE/XUKE)+(X4L2*PC42/XUKC))
      EZC43 = 2*DELX*DELY/((XDELE/XUKE)+(X4L3*PC43/XUKC))
      EZC44 = 2*DELX*DELY/((XDELE/XUKE)+(X4L4*PC44/XUKC))


      ELSE
      CONTINUE
      ENDIF
C
C
C
C     EPOXY TO BOTTOM OUTER EDGE
C
      EZB = 2*XUKE*DELX*DELY/XDELE


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     GENERATE CONSTANTS FOR THE COPPER LAYERS
C
C
C
C     LEFT OF RIGHT EDGE TO OUTSIDE
C
C
C     ONE COPPER LAYER CASE
C
C
C
C

C
      IF(SELECT.EQ.'1') THEN
      CYLR11= 2*XUKC*X1L1*PC11/DELY
C
C
C     TWO COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CYLR21= 2*XUKC*X2L1*PC21/DELY
```

```
      CYLR22= 2*XUKC*X2L2*PC22/DELY
C
C
C    THREE COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CYLR31= 2*XUKC*X3L1*PC31/DELY
      CYLR32= 2*XUKC*X3L2*PC32/DELY
      CYLR33= 2*XUKC*X3L3*PC33/DELY
C
C
C    FOUR COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CYLR41= 2*XUKC*X4L1*PC41/DELY
      CYLR42= 2*XUKC*X4L2*PC42/DELY
      CYLR43= 2*XUKC*X4L3*PC43/DELY
      CYLR44= 2*XUKC*X4L4*PC44/DELY

      ELSE
      CONTINUE
      ENDIF



C    FRONT OR BACK TO OUTSIDE
C
C
C
C
C   ONE COPPER LAYER CASE
C
C
C
C
      IF(SELECT.EQ.'1') THEN
      CFXB11= 2*XUKC*DELY*X1L1*PC11/DELX
C
C
C    TWO COPPER LAYERS CASE
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CFXB21= 2*XUKC*DELY*X2L1*PC21/DELX
      CFXB22= 2*XUKC*DELY*X2L2*PC22/DELX
C
C
C    THREE COPPER LAYERS CASE
```

```
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CFXB31= 2*XUKC*DELY*X3L1*PC31/DELX
      CFXB32= 2*XUKC*DELY*X3L2*PC32/DELX
      CFXB33= 2*XUKC*DELY*X3L3*PC33/DELX
C
C
C     FOUR COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CFXB41= 2*XUKC*DELY*X4L1*PC41/DELX
      CFXB42= 2*XUKC*DELY*X4L2*PC42/DELX
      CFXB43= 2*XUKC*DELY*X4L3*PC43/DELX
      CFXB44= 2*XUKC*DELY*X4L4*PC44/DELX

      ELSE
      CONTINUE
      ENDIF

C     INNER MATRIX MOVEMENT IN THE Y DIRECTION
C
C
C      ONE COPPER LAYER CASE
C
C
C
C
      IF(SELECT.EQ.'1') THEN
      CYY11= XUKC*DELX*X1L1*PC11/DELY
C
C
C     TWO COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CYY21= XUKC*DELX*X2L1*PC21/DELY
      CYY22= XUKC*DELX*X2L2*PC22/DELY
C
C
C      THREE COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CYY31= XUKC*DELX*X3L1*PC31/DELY
      CYY32= XUKC*DELX*X3L2*PC32/DELY
```

```
      CYY33= XUKC*DELX*X3L3*PC33/DELY
C
C
C     FOUR COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CYY41= XUKC*DELX*X4L1*PC41/DELY
      CYY42= XUKC*DELX*X4L2*PC42/DELY
      CYY43= XUKC*DELX*X4L3*PC43/DELY
      CYY44= XUKC*DELX*X4L4*PC44/DELY


      ELSE
      CONTINUE
      ENDIF

C
C
C    INNER MATRIX MOVEMENT IN THE X DIRECTION
C
C
C
C     ONE COPPER LAYER CASE
C
C
C
C
      IF(SELECT.EQ.'1') THEN
      CXX11= XUKC*X1L1*PC11*DELY/DELX
C
C
C     TWO COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CXX21= XUKC*X2L1*PC21*DELY/DELX
      CXX22= XUKC*X2L2*PC22*DELY/DELX
C
C
C     THREE COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CXX31= XUKC*X3L1*PC31*DELY/DELX
      CXX32= XUKC*X3L2*PC32*DELY/DELX
      CXX33= XUKC*X3L3*PC33*DELY/DELX
C
C
C     FOUR COPPER LAYERS CASE
```

```
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CXX41= XUKC*X4L1*PC41*DELY/DELX
      CXX42= XUKC*X4L2*PC42*DELY/DELX
      CXX43= XUKC*X4L3*PC43*DELY/DELX
      CXX44= XUKC*X4L4*PC44*DELY/DELX


      ELSE
      CONTINUE
      ENDIF
C
C     COPPER TO EPOXY (OR AIR)
C
C
C
C     ONE COPPER LAYER CASE
C
C
C
      IF(SELECT.EQ.'1') THEN
      CZE11= 2*XUKC*DELX*DELY/(X1L1*PC11)
C
C
C     TWO COPPER LAYERS CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CZE21 = 2*DELX*DELY/((XDELE/XUKE)+(X2L1*PC21/XUKC))
      CZE22 = 2*XUKC*DELX*DELY/(X2L2*PC22)
C
C
C     THREE COPPER LAYER CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CZE31 = 2*DELX*DELY/((XDELE/XUKE)+(X3L1*PC31/XUKC))
      CZE32 = 2*DELX*DELY/((XDELE/XUKE)+(X3L2*PC32/XUKC))
      CZE33 = 2*XUKC*DELX*DELY/(PC33*X3L3)
C
C
C     FOUR COPPER LAYER CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CZE41 = 2*DELX*DELY/((XDELE/XUKE)+(X4L1*PC41/XUKC))
```

```
      CZE42 = 2*DELX*DELY/((XDELE/XUKE)+(X4L2*PC42/XUKC))
      CZE43 = 2*DELX*DELY/((XDELE/XUKE)+(X4L3*PC43/XUKC))
      CZE44 = 2*XUKC*DELX*DELY/(PC44*X4L4)

      ELSE
      CONTINUE
      ENDIF
C
C     CALL THE OUTPUT DATA FILE SUBROUTINES
C
C
C
C
C
C
C      ONE COPPER LAYER CASE
C
C
C
C
      IF(SELECT.EQ.'1') THEN
      CALL S1(EYLR,EFXB,EYY,EXX,EZC11,EZB,CYLR11,CFXB11,CYY11,CXX11,CZE1
     +1,XPL,XIH,XJH,XEAT,XWIDE,XDEEP,XOEF,XIBT,XPRT,XWRT,XRT,XLT,XFT,XBT
     +,USEL)
C
C
C      TWO COPPER LAYER CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'2') THEN
      CALL S2(EYLR,EFXB,EYY,EXX,EZC21,EZC22,EZB,CYLR21,CYLR22,CFXB21,CFX
     +B22,CYY21,CYY22,CXX21,CXX22,CZE21,CZE22,XPL,XIH,XJH,XEAT,XWIDE,XDE
     +EP,XOEF,XIBT,XPRT,XWRT,XRT,XLT,XFT,XBT,USEL)
C
C
C      THREE COPPER LAYER CASE
C
C
C
      ELSEIF(SELECT.EQ.'3') THEN
      CALL S3(EYLR,EFXB,EYY,EXX,EZC31,EZC32,EZC33,EZB,CYLR31,CYLR32,CYLR
     +33,CFXB31,CFXB32,CFXB33,CYY31,CYY32,CYY33,CXX31,CXX32,CXX33,CZE31,
     +CZE32,CZE33,XPL,XIH,XJH,XEAT,XWIDE,XDEEP,XOEF,XIBT,XPRT,XWRT,XRT,X
     +LT,XFT,XBT,USEL)
C
C
C      FOUR COPPER LAYER CASE
C
C
C
C
      ELSEIF(SELECT.EQ.'4') THEN
      CALL S4(EYLR,EFXB,EYY,EXX,EZC41,EZC42,EZC43,EZC44,EZB,CYLR41,CYLR4
     +2,CYLR43,CYLR44,CFXB41,CFXB42,CFXB43,CFXB44,CYY41,CYY42,CYY43,CYY4
```

```
     +4,CXX41,CXX42,CXX43,CXX44,CZE41,CZE42,CZE43,CZE44,XPL,XIH,XJH,XEAT
     +,XWIDE,XDEEP,XOEF,XIBT,XPRT,XWRT,XRT,XLT,XFT,XBT,USEL)
      ELSE
      CONTINUE
      ENDIF
      END
      SUBROUTINE COPPER(EW,EL,ANSN,UK,SELECT,T1L1,T2L1,T2L2,T3L1,T3L2,T3
     +L3,T4L1,T4L2,T4L3,T4L4,SPEVAL,UKC,CL,CW)

C
C     TITLE:      MODEL BUILDER
C     AUTHOR:     LT STEVE GLASER
C     DATE:       09 JUL 1991
C     COMPILER:   MICROSOFT VERSION 4.01
C     LINKER:     MICROSOFT VERSION 3.55
C
C     DEFINE REAL VARIABLES
C
C     SUBROUTINE COPPER GATHERS ALL THE INFORMATION REQUIRED FOR THE COPPER
C     LAYERS.
      REAL EL,EW,UN,T1L1,T2L1,T2L2,T3L1,T3L2,T3L3,T4L1,T4L2,T4L3,T4L4,VO
     +L1,VOL2,VOL3,VOL4,T1W1,T2W1,T2W2,T3W1,T3W2,T3W3,T4W1,T4W2,T4W3,T4W
     +4,UKC

      INTEGER NWIDE,NDEEP,NPL
C     DEFINE ALL ONE CHARACTER VARIABLES
C
      CHARACTER*1 ANSN,SELECT,WORL,SPEVAL,ANSE1,ANST1,ANSK1

C     CHARACTER VARIABLES OF MORE THAN ONE POSITION

     CHARACTER LOCO*2,UK*10,UT*1
C

C     DEFINE MATRICES
C
C
C
C
C
C       COPPER LAYER CHARACTERISTICS
C
C
5458  CALL CLS
31171 WRITE(*,18000)
18000 FORMAT(///,'************************************************************
     +*********************',/,
     +'********************COPPER LAYER CHARACTERISTICS*********************
     +*********',/,
     +'************************************************************************
     +*********',/,)
C
C     PROVIDE CORRECT UNIT ABBREVIATIONS
C

        AREA=EW*EL
```

```
      IF(ANSN.EQ.'S') THEN
        UN=12*2.54*12*2.54*.00134374*2.54
        WRITE(*,15400)
15400   FORMAT('  ALL ENTRIES ARE IN SI NOTATION.',/)
         LOCO='cm'
C        UK='Watts/cm/C'
C        UT='C'
      ELSEIF(ANSN.EQ.'E') THEN
        UN=12*12*0.00134374
        WRITE(*,15401)
15401    FORMAT('  ALL ENTRIES ARE IN ENGLISH NOTATION',/)
         LOCO='in'
C        UK='Btu/hr/F'
C        UT='F'
      ENDIF
C
C
C
      WRITE(*,15402) EL,LOCO
15402 FORMAT(/,'  THE COPPER LAYER LENGTH IS THE SAME AS THE EPOXY LAYER
     +:',F9.4,1X,A2,2X, )
      CL=EL
C
      WRITE(*,15403) EW,LOCO
15403 FORMAT(//,'  THE COPPER LAYER WIDTH IS THE SAME AS THE EPOXY LAYER
     +: ',F9.4,1X,A2,2X, )
      CW=EW
14337 WRITE(*,16403) SELECT
16403 FORMAT(//,'  YOU SELECTED ',A1,' COPPER LAYER(S) FOR THE PCB.',/,'
     +  YOU NOW WILL BE ASKED TO ENTER THE COPPER LAYER THICKNESS',2X, )
99999 WRITE(*,31112)
31112 FORMAT(//,'  DO YOU WANT TO SPECIFY THICKNESS BY LENGTH OR WEIGHT?
     +',/,'  ENTER L IF YOU WANT TO ENTER LENGTH, W IF YOU WANT TO ENTER
     + WEIGHT: ',2X, )
      READ(*,31113) WORL
31113 FORMAT(A1)
      IF(WORL.EQ.'L'.AND.SELECT.EQ.'1') THEN
        CALL CLS
        WRITE(*,31114) LOCO
31114   FORMAT(////,'  ENTER THE THICKNESS FOR LAYER 1 (',A2,'): ',2X, )
        READ *,T1L1
        GOTO 31170
      ELSEIF(WORL.EQ.'L'.AND.SELECT.EQ.'2')THEN
        CALL CLS
        WRITE(*,31115) LOCO
31115   FORMAT(////,'  ENTER THE THICKNESS FOR LAYER 1 (',A2,'): ',2X, )
        READ *,T2L1
        WRITE(*,31116) LOCO
31116   FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 2 (',A2,'): ',2X, )
        READ *,T2L2
        GOTO 31170
         ELSEIF(WORL.EQ.'L'.AND.SELECT.EQ.'3')THEN
        CALL CLS
        WRITE(*,31117) LOCO
31117   FORMAT(////,'  ENTER THE THICKNESS FOR LAYER 1 (',A2,'): ',2X, )
        READ *,T3L1
```

```
      WRITE(*,31118) LOCO
31118 FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 2 (',A2,'): ',2X, )
      READ *,T3L2
      WRITE(*,31119) LOCO
31119 FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 3 (',A2,'): ',2X, )
      READ *,T3L3
      GOTO 31170
       ELSEIF(WORL.EQ.'L'.AND.SELECT.EQ.'4')THEN
      CALL CLS
      WRITE(*,31120) LOCO
31120 FORMAT(////,'  ENTER THE THICKNESS FOR LAYER 1 (',A2,'): ',2X, )
      READ *,T4L1
      WRITE(*,31121) LOCO
31121 FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 2 (',A2,'): ',2X, )
      READ *,T4L2
      WRITE(*,31122) LOCO
31122 FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 3 (',A2,'): ',2X, )
      READ *,T4L3
      WRITE(*,31123) LOCO
31123 FORMAT(/,'  ENTER THE THICKNESS FOR LAYER 4 (',A2,'): ',2X, )
      READ *,T4L4
      GOTO 31170
      ELSEIF(WORL.EQ.'W'.AND.SELECT.EQ.'1') THEN
      CALL CLS
      WRITE(*,31142)
31142 FORMAT(////,'  ENTER THE WEIGHT FOR LAYER 1 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T1W1
      T1W1=T1W1*62.5
      ELSE
      READ *,T1W1
      ENDIF
      VOL1=UN*T1W1
      T1L1=VOL1/AREA
      GOTO 31170
       ELSEIF(WORL.EQ.'W'.AND.SELECT.EQ.'2')THEN
      CALL CLS
      WRITE(*,31143)
31143 FORMAT(////,'  ENTER THE WEIGHT FOR LAYER 1 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T2W1
      T2W1=T2W1*62.5
      ELSE
      READ *,T2W1
      ENDIF
      VOL1=UN*T2W1
      T2L1=VOL1/AREA
      WRITE(*,31144)
31144 FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 2 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T2W2
      T2W2=T2W2*62.5
      ELSE
      READ *,T2W2
      ENDIF
      VOL2=UN*T2W2
```

```
      T2L2=VOL2/AREA
    GOTO 31170
       ELSEIF(WORL.EQ.'W'.AND.SELECT.EQ.'3')THEN
      CALL CLS
      WRITE(*,31145)
31145 FORMAT(////,'  ENTER THE WEIGHT FOR LAYER 1 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T3W1
      T3W1=T3W1*62.5
      ELSE
      READ *,T3W1
      ENDIF
      VOL1=UN*T3W1
      T3L1=VOL1/AREA
       WRITE(*,31146)
31146 FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 2 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T3W2
      T3W2=T3W2*62.5
      ELSE
      READ *,T3W2
      ENDIF
      VOL2=UN*T3W2
      T3L2=VOL2/AREA
      WRITE(*,31147)
31147 FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 3 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T3W3
      T3W3=T3W3*62.5
      ELSE
      READ *,T3W3
      ENDIF
      VOL3=UN*T3W3
      T3L3=VOL3/AREA
    GOTO 31170
       ELSEIF(WORL.EQ.'W'.AND.SELECT.EQ.'4')THEN
      CALL CLS
      WRITE(*,31148)
31148 FORMAT(////,'  ENTER THE WEIGHT FOR LAYER 1 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T4W1
      T4W1=T4W1*62.5
      ELSE
      READ *,T4W1
      ENDIF
      VOL1=UN*T4W1
      T4L1=VOL1/AREA
    WRITE(*,31149)
31149 FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 2 (oz): ',2X, )
      IF(UW.EQ.'gm') THEN
      READ *,T4W2
      T4W2=T4W2*62.5
      ELSE
      READ *,T4W2
      ENDIF
      VOL2=UN*T4W2
```

```
        T4L2=VOL2/AREA
      WRITE(*,31150)
31150   FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 3 (oz): ',2X, )
        IF(UW.EQ.'gm') THEN
        READ *,T4W3
        T4W3=T4W3*62.5
        ELSE
        READ *,T4W3
        ENDIF
        VOL3=UN*T4W3
        T4L3=VOL3/AREA
      WRITE(*,31151)
31151   FORMAT(/,'  ENTER THE WEIGHT FOR LAYER 4 (oz): ',2X, )
        IF(UW.EQ.'gm') THEN
        READ *,T4W4
        T4W4=T4W4*62.5
        ELSE
        READ *,T4W4
        ENDIF
        VOL4=UN*T4W4
        T4L4=VOL4/AREA
      GOTO 31170
        CALL CLS
        GOTO 31170
        ELSE
        CALL CLS
        GOTO 31171
      ENDIF
31170 IF(SPEVAL.EQ.'B') THEN
      GOTO 41186
      ELSE
      CONTINUE
      ENDIF
      WRITE(*,31172) UK
31172 FORMAT(/,'  ENTER COPPER LAYER THERMAL CONDUCTIVITY (',A10,'): ',
     +2X, )
      READ *,UKC
C
C    MAKE CHANGES OR CORRECTIONS TO COPPER LAYER ENTRIES
C
31193 IF(SELECT.EQ.'1')THEN
      CALL CLS
      WRITE(*,31173) CL,LOCO
31173 FORMAT(///,'      YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE COPPE
     +R LAYER(S).',///,
     +'      1.) LENGTH:                 ',F9.4,1X,A2,2X, )
      WRITE(*,73) CW,LOCO
73    FORMAT(/,'      2.) WIDTH:                 ',F9.4,1X,A2,2X, )
      WRITE(*,74) T1L1,LOCO
74    FORMAT(/,'      3.) THICKNESS LAYER 1: ',F9.4,1X,A2,2X, )
      WRITE(*,75) UKC,UK
75    FORMAT(/,'      4.) k:                     ',F9.4,1X,A10,2X, )
      ELSEIF(SELECT.EQ.'2')THEN
      CALL CLS
      WRITE(*,311) CL,LOCO
311   FORMAT(///,'      YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE COPPE
```

```
     +R LAYER(S).',///,
     +'      1.) LENGTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,312) CW,LOCO
312   FORMAT(/,'      2.) WIDTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,314) T2L1,LOCO
314   FORMAT(/,'      3.) THICKNESS LAYER 1: ',F9.4,1X,A2,2X, )
      WRITE(*,315) T2L2,LOCO
315   FORMAT(/,'          THICKNESS LAYER 2: ',F9.4,1X,A2,2X, )
      WRITE(*,316) UKC,UK
316   FORMAT(/,'      4.) k:                  ',F9.4,1X,A10,2X, )
      ELSEIF(SELECT.EQ.'3')THEN
      CALL CLS
      WRITE(*,317) CL,LOCO
317   FORMAT(///,'      YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE COPPE
     +R LAYER(S).',///,
     +'      1.) LENGTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,318) CW,LOCO
318   FORMAT(/,'      2.) WIDTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,319) T3L1,LOCO
319   FORMAT(/,'      3.) THICKNESS LAYER 1: ',F9.4,1X,A2,2X, )
      WRITE(*,320) T3L2,LOCO
320   FORMAT(/,'          THICKNESS LAYER 2: ',F9.4,1X,A2,2X, )
      WRITE(*,321) T3L3,LOCO
321   FORMAT(/,'          THICKNESS LAYER 3: ',F9.4,1X,A2,2X, )
      WRITE(*,322) UKC,UK
322   FORMAT(/,'      4.) k:                  ',F9.4,1X,A10,2X, )
      ELSEIF(SELECT.EQ.'4')THEN
      CALL CLS
      WRITE(*,323) CL,LOCO
323   FORMAT(///,'      YOU HAVE MADE THE FOLLOWING ENTRIES FOR THE COPPE
     +R LAYER(S).',///,
     +'      1.) LENGTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,324) CW,LOCO
324   FORMAT(/,'      2.) WIDTH:              ',F9.4,1X,A2,2X, )
      WRITE(*,325) T4L1,LOCO
325   FORMAT(/,'      3.) THICKNESS LAYER 1: ',F9.4,1X,A2,2X, )
      WRITE(*,326) T4L2,LOCO
326   FORMAT(/,'          THICKNESS LAYER 2: ',F9.4,1X,A2,2X, )
      WRITE(*,327) T4L3,LOCO
327   FORMAT(/,'          THICKNESS LAYER 3: ',F9.4,1X,A2,2X, )
      WRITE(*,328) T4L4,LOCO
328   FORMAT(/,'          THICKNESS LAYER 4: ',F9.4,1X,A2,2X, )
      WRITE(*,329) UKC,UK
329   FORMAT(/,'      4.) k:                  ',F9.4,1X,A10,2X, )
      ENDIF
C
31194 WRITE(*,71180)
71180 FORMAT(///,'      DO YOU WISH TO MAKE ANY CHANGES?  SELECT Y FOR Y
     +ES AND N FOR NO: ',2X, )
      READ(*,31181)ANSE1
31181 FORMAT(A1)
      IF(ANSE1.EQ.'Y') THEN
56342    WRITE(*,31184)
31184    FORMAT(/,'      WOULD YOU LIKE TO CHANGE THE THICKNESS?  (Y OR N
     +): ',2X, )
```

```
        READ(*,31185)ANST1
31185   FORMAT(A1)
        PRINT *
          IF(ANST1.EQ.'Y') THEN
          SPEVAL='B'
          CALL CLS
          WRITE(*,99998)
99998     FORMAT(////)
          GOTO 99999
          ELSEIF(ANST1.EQ.'N') THEN
          CONTINUE
          ELSE
          CALL CLS
          WRITE(*,56343)
56343     FORMAT(////)
          GOTO 56342
          ENDIF
C
41186      CALL CLS
          WRITE(*,31187)
31187 FORMAT(////)
        PRINT *,'    THE CURRENT ENTRY FOR THERMAL CONDUCTIVITY IS ',UK
     +C,' ',UK
32186   WRITE(*,31188)
31188   FORMAT(/,'     WOULD YOU LIKE TO CHANGE THE THERMAL CONDUCTIVIT
     +Y?  (Y OR N):  ',2X, )
        READ(*,31189)ANSK1
31189    FORMAT(A1)
        PRINT *
          IF(ANSK1.EQ.'Y') THEN
          WRITE(*,31190) UK
31190  FORMAT(/,'     ENTER THE THERMAL CONDUCTIVITY (',A10,'):  ',2X, )
          READ *,UKC
          ELSEIF(ANSK1.EQ.'N') THEN
          GOTO 31191
          ELSE
          CALL CLS
          WRITE(*,56344)
56344     FORMAT(////)
          GOTO 32186
          ENDIF
31191     GOTO 31193
C
C
      ELSEIF(ANSE1.EQ.'N') THEN
      GOTO 31180
      ELSE
      CALL CLS
      GOTO 31194
      ENDIF
31180 END
C*********************************************************************
C
C    TITLE:      MODEL BUILDER
C    SUBROUTINE: S1
C    DATE:       09 JUL 91
```

```
C     AUTHOR:       LT STEVE GLASER
C     COMPILER:     MICROSOFT VERSION 4.01
C     LINKER:       MICROSOFT VERSION 3.55
C
C     CALLED FROM SUBROUTINE PCBS3. THIS SUBROUTINE GENERATES THE OUTPUT
C     DATA FILE FOR THE ONE COPPER LAYER CASE
C
C
      SUBROUTINE S1 (GYLR,GFXB,GYY,GXX,GZC11,GZB,GYLR11,GFXB11,GYY11,GXX
     +11,GZE11,GPL,GIH,GJH,GEAT,GWIDE,GDEEP,GOEF,GIBT,GPRT,GWRT,GRT,GLT,
     +GFT,GBT,GSEL)

      REAL GYLR,GFXB,GYY,GXX,GZC11,GZB,GYLR11,GFXB11,GYY11,GXX11,GZE11,A
     +CC,DAMP,CONFAC,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT
      REAL GEAT(100,100),GOEF(1000,12)


      CHARACTER*1 SELECT,ANS
      CHARACTER DATAF*79,NAME*6

      INTEGER GPL,I,N,IB,GWIDE,GDEEP,COUNT,CONTEMP,ZER,GSEL,NMAX,TMAX,HT
     +RS,D1,D2,D3,D4,D5,D6,D7,MAXIT,LOCVAR

      INTEGER GIH(1000),GJH(1000),GCON(1000,12)
      CONTEMP=6
      ZER=0
      NMAX=750
      TMAX=50
      HTRS=6
      D1=2
      D2=4
      D3=6
      D4=0
      D5=0
      D6=0
      D7=0
      ACC=0.05
      DAMP=0.666667
      MAXIT=12
      CONFAC=0.8
      COUNT=2*GPL
      LOCVAR=COUNT


C
C
898   CALL CLS

      WRITE(*,803)
803   FORMAT(///,'     THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY
     + INTO THE',/,
     +'     EXISTING THERMAL ANALYZER, FURTHERMORE, THIS PROGRAM DOES',/,
     +'     NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE',/,
     +'     THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS     ',/,
     +'     PROGRAM.  THE FILE NAME IS LIMITED TO SIX CHARACTERS, AND',/,
     +'     SHOULD NOT HAVE ANY SPACES.                               ',///,
```

```
     +'     PLEASE ENTER THE DESIRED DATA FILE NAME:   ',2X )
      READ(*,804) NAME
804   FORMAT(A6)
552   WRITE(*,910)NAME
910   FORMAT(/////,'    YOU SELECTED ',A6,' FOR YOUR DATA FILE NAME
     +',//)
911   WRITE(*,811)
811   FORMAT('     IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N
     +FOR NO:    ',2X, )
      READ(*,812) ANS
812   FORMAT(A1)
C
      IF(ANS.EQ.'N') THEN
        GOTO 898
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'Y') THEN
        GOTO 897
      ELSE
         CALL CLS
         GOTO 552
897     ENDIF

C     ALLOW THE USER TO PROVIDE TITLE LINE FOR DATA FILE

833   CALL CLS
      WRITE(*,805)
805   FORMAT(////,'          ENTER THE DESIRED TITLE TO BE PLACED ON LINE
     +',/,'          NUMBER ONE OF THE OUTPUT DATA FILE:',///,
     +'           ',2X, )
      READ(*,806) DATAF
806   FORMAT(A79)
835   WRITE(*,831)
831   FORMAT(////,'          DO YOU WISH TO CHANGE THE TITLE OF YOUR OUTPU
     +T DATA FILE?',/,'          ENTER Y FOR YES AND N FOR NO:    ',2X, )
      READ(*,832) ANS
832   FORMAT(A1)
C
      IF(ANS.EQ.'Y') THEN
        GOTO 833
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'N') THEN
        GOTO 834
      ELSE
        GOTO 835
834     ENDIF

C************************************************************************
C**************COEFFICIENTS FOR EPOXY AND COPPER LAYERS*****************
C************************************************************************

C
      DO 90 I=1,GPL
```

```
      N=1
      IB=GPL+I
C
C
C*********************************CORNERS*********************************
C
      IF ((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I).EQ
     +.GWIDE)) THEN
C
C    DETERMINE COEFFICIENTS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C    CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
C
C
C    LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C    LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C    LEFT COEFFICIENT
      GOEF(I,N) = GYLR11
      GOEF(IB,N) = GYLR
      N=N+1
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
C
C    RIGHT COEFFICIENT
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
C
C    RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C    LEFT COEFFICIENT
C
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
C
C    RIGHT COEFFICIENT
C
      GOEF(I,N) = GYLR11
      GOEF(IB,N) = GYLR
      N=N+1
      GCON(I,N) = 7541
      GCON(IB,N) = 7541
```

```
      ENDIF
C
C     FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GFXB11
      GOEF(IB,N)=GFXB
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521

C     BACK COEFFICIENT
      GOEF(I,N)=GXX11
      GOEF(IB,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1

C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX11
      GOEF(IB,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1

C     BACK COEFFICIENT
      GOEF(I,N)=GFXB11
      GOEF(IB,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      ENDIF
C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE11
      GOEF(IB,N)=GZC11
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC11
      GOEF(IB,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
```

```
      GCON(I,N)=9991
      ENDIF
C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE NUMBER OF CONNECTIONS FOR COPPER AND EPOXY LAYERS
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
C
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
C
C
C     FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GFXB11
      GOEF(IB,N)=GFXB
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521

C     BACK COEFFICIENT
      GOEF(I,N)=GXX11
      GOEF(IB,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1

C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX11
```

```
      GOEF(IB,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1

C     BACK COEFFICIENT
      GOEF(I,N)=GFXB11
      GOEF(IB,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      ENDIF
C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE11
      GOEF(IB,N)=GZC11
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC11
      GOEF(IB,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF

CXCXCXC
C
C
C     LEFT AND RIGHT EDGES EXCLUDING CORNERS
C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I)
     +.EQ.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
```

```
C
C     LEFT EDGE
       IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
       GOEF(I,N) = GYLR11
       GOEF(IB,N) = GYLR
       N=N+1
       GCON(I,N) = 7551
       GCON(IB,N) = 7551
C
C     RIGHT COEFFICIENT
       GOEF(I,N) = GYY11
       GOEF(IB,N) = GYY
       N=N+1
       GCON(I,N) = 10 * (I+1) + 1
       GCON(IB,N) = 10* (IB+1) + 1
C
C     RIGHT EDGE
       ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C     LEFT COEFFICIENT
C
       GOEF(I,N) = GYY11
       GOEF(IB,N) = GYY
       N=N+1
       GCON(I,N) = 10*(I-1)+1
       GCON(IB,N) = 10*(IB-1)+1
C
C     RIGHT COEFFICIENT
C
       GOEF(I,N) = GYLR11
       GOEF(IB,N) = GYLR
       N=N+1
       GCON(I,N) = 7541
       GCON(IB,N) = 7541
       ENDIF
C
C
C     FRONT COEFFICIENT
       GOEF(I,N)=GXX11
       GOEF(IB,N)=GXX
       N=N+1

       GCON(I,N) =10*(I-GWIDE)+1
       GCON(IB,N) = 10*(IB-GWIDE)+1

C     BACK COEFFICIENT
       GOEF(I,N)=GXX11
       GOEF(IB,N)=GXX
       N=N+1
       GCON(I,N) =10*(I+GWIDE)+1
       GCON(IB,N) = 10*(IB+GWIDE)+1

C
C     TOP COEFFICIENT
       GOEF(I,N)=GZE11
```

```
      GOEF(IB,N)=GZC11
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC11
      GOEF(IB,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
C
C
C
C****DETERMINE COEFFICIENTS FOR ALL NODES NOT TOUCHING AN EDGE*******
C
      ELSEIF((GIH(I).NE.1.OR.GIH(I).NE.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE CONNECTIONS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
C
C
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1



C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY11
      GOEF(IB,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
C
C
```

```
C
C     FRONT COEFFICIENT
       GOEF(I,N)=GXX11
       GOEF(IB,N)=GXX
       N=N+1
       GCON(I,N) =10*(I-GWIDE)+1
       GCON(IB,N) = 10*(IB-GWIDE)+1

C     BACK COEFFICIENT
       GOEF(I,N)=GXX11
       GOEF(IB,N)=GXX
       N=N+1
       GCON(I,N) =10*(I+GWIDE)+1
       GCON(IB,N) = 10*(IB+GWIDE)+1

C
C     TOP COEFFICIENT
       GOEF(I,N)=GZE11
       GOEF(IB,N)=GZC11
       N=N+1
       GCON(I,N) = 7511
       GCON(IB,N) = 10*I+1
C
C     BOTTOM COEFFICIENT
C
       GOEF(I,N) =  GZC11
       GOEF(IB,N) = GZB
       N=N+1
       GCON(I,N) =10*(I+GPL)+1
       GCON(IB,N) = 10*(IB+GPL)+1
C
C     HEAT INPUT
       IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
       GOEF(I,N) = GEAT(GIH(I),GJH(I))
       N=N+1
       GCON(I,N)=9991
       ENDIF
       ENDIF
90     CONTINUE
C
C      GENERATE DATA FILE
C
       OPEN (3,FILE=NAME,FORM='FORMATTED',ACCESS='DIRECT',RECL=108,STATUS
     +='NEW')
       WRITE(3,909) DATAF
909    FORMAT(1X,A79)
       WRITE(3,908) COUNT,CONTEMP,ZER,ZER,ZER,ZER,ZER,ZER,GSEL
908    FORMAT(2X,9(I3,5X))
       WRITE(3,907) ZER,ZER,ZER
907    FORMAT(2X,3(I3,5X))
       WRITE(3,9081) NMAX,TMAX,HTRS,D1,D2,D3,D4,D5,D6,D7
9081   FORMAT(2X,9(I3,5X))
       WRITE(3,905) ACC,DAMP,MAXIT,CONFAC,GIBT
905    FORMAT(1X,2(F9.7,1X),14X,I2,1X,F9.7,1X,F9.5)
       WRITE(3,906) GPRT,GFT,GBT,GRT,GLT,GWRT
906    FORMAT(1X,6(F12.4,1X))
```

```
      DO 112 I=1,LOCVAR
      WRITE(3,9100) GCON(I,1),GCON(I,2),GCON(I,3),GCON(I,4),GCON(I,5),GC
     +ON(I,6),GCON(I,7),GCON(I,8)
9100  FORMAT(I4,3X,7(I5,7X))
      WRITE(3,9110) GOEF(I,1),GOEF(I,2),GOEF(I,3),GOEF(I,4),GOEF(I,5),GO
     +EF(I,6),GOEF(I,7)
9110  FORMAT(7(F9.3,3X))
112   CONTINUE
      CLOSE (3)
      CALL CLS
      WRITE(*,999) NAME
999   FORMAT(////,'          THE OUTPUT DATA HAS BEEN PLACED IN A FILE
     +NAMED ',A6,/////,'                         <PRESS ENTER TO CONTINUE>')
      READ(*,5912) ANS
5912  FORMAT(A1)

      END
C***********************************************************************
C
C     TITLE:       MODEL BUILDER
C     SUBROUTINE:  S2
C     DATE:        09 JUL 91
C     AUTHOR:      LT STEVE GLASER
C     COMPILER:    MICROSOFT VERSION 4.01
C     LINKER:      MICROSOFT VERSION 3.55
C
C     S2 IS CALLED FROM PCBS3.  S2 IS THE OUTPUT DATA FILE GENERATOR FOR THE
C     THE COPPER LAYER CASE.
C
      SUBROUTINE S2 (GYLR,GFXB,GYY,GXX,GZC21,GZC22,GZB,GYLR21,GYLR22,GFX
     +B21,GFXB22,GYY21,GYY22,GXX21,GXX22,GZE21,GZE22,GPL,GIH,GJH,GEAT,GW
     +IDE,GDEEP,GOEF,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT,GSEL)

      REAL GYLR,GFXB,GYY,GXX,GZC21,GZB,GYLR21,GFXB21,GYY21,GXX21,GZE21,A
     +CC,DAMP,CONFAC,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT,GYLR22,GFXB22,GYY22,
     +GXX22,GZE22,GZC22

      REAL GEAT(100,100),GOEF(1000,12)


      CHARACTER*1 SELECT,ANS
      CHARACTER DATAF*79,NAME*6

      INTEGER GPL,I,N,IB,GWIDE,GDEEP,COUNT,CONTEMP,ZER,GSEL,NMAX,TMAX,HT
     +RS,D1,D2,D3,D4,D5,D6,D7,MAXIT,LOCVAR,IC,ID

      INTEGER GIH(1000),GJH(1000),GCON(1000,12)
      CONTEMP=6
      ZER=0
      NMAX=750
      TMAX=50
      HTRS=6
      D1=2
      D2=4
      D3=6
      D4=0
```

```
      D5=0
      D6=0
      D7=0
      ACC=0.05
      DAMP=0.666667
      MAXIT=12
      CONFAC=0.8
      COUNT=4*GPL
      LOCVAR=COUNT


C
C
898   CALL CLS

      WRITE(*,803)
803   FORMAT(///,'     THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY
     + INTO THE',/,
     +'    EXISTING THERMAL ANALYZER, FURTHERMORE, THIS PROGRAM DOES',/,
     +'    NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE',/,
     +'    THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS    ',/,
     +'    PROGRAM.  THE FILE NAME IS LIMITED TO SIX CHARACTERS, AND',/,
     +'    SHOULD NOT HAVE ANY SPACES.                             ',///,
     +'    PLEASE ENTER THE DESIRED DATA FILE NAME:  ',2X )
      READ(*,804) NAME
804   FORMAT(A6)
552   WRITE(*,910)NAME
910   FORMAT(/////,'    YOU SELECTED ',A6,' FOR YOUR DATA FILE NAME
     +',//)
911   WRITE(*,811)
811   FORMAT('     IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N
     +FOR NO:   ',2X, )
      READ(*,812) ANS
812   FORMAT(A1)
C
      IF(ANS.EQ.'N') THEN
        GOTO 898
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'Y') THEN
        GOTO 897
      ELSE
         CALL CLS
         GOTO 552
897     ENDIF

C     ALLOW THE USER TO PROVIDE TITLE LINE FOR DATA FILE

833   CALL CLS
      WRITE(*,805)
805   FORMAT(////,'         ENTER THE DESIRED TITLE TO BE PLACED ON LINE
     +',/,'        NUMBER ONE OF THE OUTPUT DATA FILE:',///,
     +'          ',2X, )
      READ(*,806) DATAF
806   FORMAT(A79)
```

```
835   WRITE(*,831)
831   FORMAT(////,'        DO YOU WISH TO CHANGE THE TITLE OF YOUR OUTPU
     +T DATA FILE?',/,'        ENTER Y FOR YES AND N FOR NO:   ',2X, )
      READ(*,832) ANS
832   FORMAT(A1)
C
      IF(ANS.EQ.'Y') THEN
        GOTO 833
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'N') THEN
        GOTO 834
      ELSE
        GOTO 835
834     ENDIF

C************************************************************************
C**************COEFFICIENTS FOR EPOXY AND COPPER LAYERS*****************
C************************************************************************

C
      DO 90 I=1,GPL
      N=1
      IB=GPL+I
      IC=2*GPL+I
      ID=3*GPL+I
C
C**********************************CORNERS*********************************
C
      IF ((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I).EQ
     +.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP COPPER
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(ID,N) = 6
      GCON(IC,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
      GOEF(I,N) = GYLR21
      GOEF(IC,N) = GYLR22
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      N=N+1
```

```
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
      GCON(IC,N) = 7551
      GCON(ID,N) = 7551
C
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY21
      GOEF(IC,N) = GYY22
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
C
C     RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C     LEFT COEFFICIENT
C
      GOEF(I,N) = GYY21
      GOEF(IC,N) = GYY22
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY

      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1


C
C     RIGHT COEFFICIENT
C
      GOEF(I,N) = GYLR21
      GOEF(IC,N) = GYLR22
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      N=N+1
      GCON(I,N) = 7541
      GCON(IB,N) = 7541
      GCON(IC,N) = 7541
      GCON(ID,N) = 7541
      ENDIF
C
C     FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GFXB21
      GOEF(IC,N)=GFXB22
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
```

```
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521

C     BACK COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1


C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1

C     BACK COEFFICIENT
      GOEF(I,N)=GFXB21
      GOEF(IC,N)=GFXB22
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      GCON(IC,N) = 7511
      GCON(ID,N) = 7511

      ENDIF
C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE21
      GOEF(IC,N)=GZE22
      GOEF(IB,N)=GZC21
      GOEF(ID,N)=GZC22
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
C
```

```
C     BOTTOM COEFFICIENT
C
       GOEF(I,N) =  GZC21
       GOEF(IC,N) = GZC22
       GOEF(IB,N) = GZE22
       GOEF(ID,N) = GZB
       N=N+1
       GCON(I,N) =10*(I+GPL)+1
       GCON(IB,N) = 10*(IB+GPL)+1
       GCON(IC,N) = 10*(IC+GPL)+1
       GCON(ID,N) = 10*(ID+GPL)+1


C
C     HEAT INPUT
       IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
       GOEF(I,N) = GEAT(GIH(I),GJH(I))
       N=N+1
       GCON(I,N)=9991
       ENDIF

C
       ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE NUMBER OF CONNECTIONS FOR COPPER AND EPOXY LAYERS
       IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
           GCON(I,N) = 6
       ELSE
           GCON(I,N) = 7
       ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
       GCON(IB,N) = 6
       GCON(ID,N) = 6
       GCON(IC,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS
C     LEFT COEFFICIENT
       GOEF(I,N) = GYY21
       GOEF(IC,N) = GYY22
       GOEF(IB,N) = GYY
       GOEF(ID,N) = GYY
       N=N+1
       GCON(I,N) = 10*(I-1)+1
       GCON(IB,N) = 10*(IB-1)+1
       GCON(IC,N) = 10*(IC-1)+1
       GCON(ID,N) = 10*(ID-1)+1


C
C     RIGHT COEFFICIENT
       GOEF(I,N) = GYY21
       GOEF(IC,N) = GYY22
       GOEF(IB,N) = GYY
```

```
      GOEF(ID,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
C
C
C     FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GFXB21
      GOEF(IC,N)=GFXB22
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521

C     BACK COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1

C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1

C     BACK COEFFICIENT
      GOEF(I,N)=GFXB21
      GOEF(IC,N)=GFXB22
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      GCON(IC,N) = 7511
```

```
      GCON(ID,N) = 7511
      ENDIF
C
C    TOP COEFFICIENT
      GOEF(I,N)=GZE21
      GOEF(IC,N)=GZE22
      GOEF(IB,N)=GZC21
      GOEF(ID,N)=GZC22
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
C
C    BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC21
      GOEF(IC,N) = GZC22
      GOEF(IB,N) = GZB
      GOEF(ID,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
C
C    HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF

CXCXCXC
CXCXCXC
C
C
C    LEFT AND RIGHT EDGES EXCLUDING CORNERS
C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I)
     +.EQ.GWIDE)) THEN
C
C    DETERMINE COEFFICIENTS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C    CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6

C
C
```

```
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     LEFT EDGE
       IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
       GOEF(I,N) = GYLR21
       GOEF(IC,N) = GYLR22
       GOEF(IB,N) = GYLR
       GOEF(ID,N) = GYLR
       N=N+1
       GCON(I,N) = 7551
       GCON(IB,N) = 7551
       GCON(IC,N) = 7551
       GCON(ID,N) = 7551
C
C     RIGHT COEFFICIENT
       GOEF(I,N) = GYY21
       GOEF(IC,N) = GYY22
       GOEF(IB,N) = GYY
       GOEF(ID,N) = GYY
       N=N+1
       GCON(I,N) = 10 * (I+1) + 1
       GCON(IB,N) = 10* (IB+1) + 1
       GCON(IC,N) = 10 * (IC+1) + 1
       GCON(ID,N) = 10 * (ID+1) + 1
C
C     RIGHT EDGE
       ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C     LEFT COEFFICIENT
       GOEF(I,N) = GYY21
       GOEF(IC,N) = GYY22
       GOEF(IB,N) = GYY
       GOEF(ID,N) = GYY

       N=N+1
       GCON(I,N) = 10*(I-1)+1
       GCON(IB,N) = 10*(IB-1)+1
       GCON(IC,N) = 10*(IC-1)+1
       GCON(ID,N) = 10*(ID-1)+1
C
C     RIGHT COEFFICIENT
C
       GOEF(I,N) = GYLR21
       GOEF(IC,N) = GYLR22
       GOEF(IB,N) = GYLR
       GOEF(ID,N) = GYLR
       N=N+1
       GCON(I,N) = 7541
       GCON(IB,N) = 7541
       GCON(IC,N) = 7541
       GCON(ID,N) = 7541
       ENDIF
C
C
C     FRONT COEFFICIENT
```

```
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1

C     BACK COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1

C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE21
      GOEF(IC,N)=GZE22
      GOEF(IB,N)=GZC21
      GOEF(ID,N)=GZC22
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC21
      GOEF(IC,N) = GZC22
      GOEF(IB,N) = GZE22
      GOEF(ID,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
C
C
C
C****DETERMINE COEFFICIENTS FOR ALL NODES NOT TOUCHING AN EDGE*******
C
```

```
      ELSEIF((GIH(I).NE.1.OR.GIH(I).NE.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE CONNECTIONS FOR TOP LAYER
       IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
           GCON(I,N) = 7
      ELSE
           GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
C
C
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY21
      GOEF(IC,N) = GYY22
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I-1) + 1
      GCON(IB,N) = 10* (IB-1) + 1
      GCON(IC,N) = 10 * (IC-1) + 1
      GCON(ID,N) = 10 * (ID-1) + 1



C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY21
      GOEF(IC,N) = GYY22
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
C
C
C
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1


C     BACK COEFFICIENT
```

```
      GOEF(I,N)=GXX21
      GOEF(IC,N)=GXX22
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1


C
C    TOP COEFFICIENT
      GOEF(I,N)=GZE21
      GOEF(IC,N)=GZE22
      GOEF(IB,N)=GZC21
      GOEF(ID,N)=GZC22
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(IC,N) = 7511
      GCON(ID,N) = 10*IC+1

C
C    BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZC21
      GOEF(IC,N) = GZC22
      GOEF(IB,N) = GZB
      GOEF(ID,N) = GZE22
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1

C
C    HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
      ENDIF
90    CONTINUE
C
C     GENERATE DATA FILE
C
      OPEN (3,FILE=NAME,FORM='FORMATTED',ACCESS='DIRECT',RECL=108,STATUS
     +='NEW')
      WRITE(3,909) DATAF
909   FORMAT(1X,A79)
      WRITE(3,908) COUNT,CONTEMP,ZER,ZER,ZER,ZER,ZER,ZER,GSEL
908   FORMAT(2X,9(I3,5X))
      WRITE(3,907) ZER,ZER,ZER
907   FORMAT(2X,3(I3,5X))
```

```
      WRITE(3,9081) NMAX,TMAX,HTRS,D1,D2,D3,D4,D5,D6,D7
9081  FORMAT(2X,9(I3,5X))
      WRITE(3,905) ACC,DAMP,MAXIT,CONFAC,GIBT
905   FORMAT(1X,2(F9.7,1X),14X,I2,1X,F9.7,1X,F9.5)
      WRITE(3,906) GPRT,GFT,GBT,GRT,GLT,GWRT
906   FORMAT(1X,6(F12.3,1X))
      DO 112 I=1,LOCVAR
      WRITE(3,9100) GCON(I,1),GCON(I,2),GCON(I,3),GCON(I,4),GCON(I,5),GC
     +ON(I,6),GCON(I,7),GCON(I,8)
9100  FORMAT(I4,3X,7(I5,7X))
      WRITE(3,9110) GOEF(I,1),GOEF(I,2),GOEF(I,3),GOEF(I,4),GOEF(I,5),GO
     +EF(I,6),GOEF(I,7)
9110  FORMAT(7(F9.3,3X))
112   CONTINUE
      CLOSE (3)
      CALL CLS
      WRITE(*,999) NAME
999   FORMAT(////,'             THE OUTPUT DATA HAS BEEN PLACED IN A FILE
     +NAMED ',A6,/////,'                            <PRESS ENTER TO CONTINUE>')
      READ(*,5912) ANS
5912  FORMAT(A1)

      END
C
C
C     TITLE:       MODEL BUILDER
C     SUBROUTINE:  S3
C     DATE:        09 JUL 91
C     AUTHOR:      LT STEVE GLASER
C     COMPILER:    MICROSOFT VERSION 4.01
C     LINKER:      MICROSOFT VERSION 3.55
C
C     SUBROUTINE S3 IS CALLED FROM SUBROUTINE PCBS3.  S3 GENERATES THE
C     OUTPUT DATA FILE FOR THE THREE COPPER LAYER CASE.
C
      SUBROUTINE S3 (GYLR,GFXB,GYY,GXX,GZC31,GZC32,GZC33,GZB,GYLR31,GYLR
     +32,GYLR33,GFXB31,GFXB32,GFXB33,GYY31,GYY32,GYY33,GXX31,GXX32,GXX33
     +,GZE31,GZE32,GZE33,GPL,GIH,GJH,GEAT,GWIDE,GDEEP,GOEF,GIBT,GPRT,GWR
     +T,GRT,GLT,GFT,GBT,GSEL)

      REAL GYLR,GFXB,GYY,GXX,GZC31,GZB,GYLR31,GFXB31,GYY31,GXX31,GZE31,A
     +CC,DAMP,CONFAC,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT,GYLR32,GFXB32,GYY32,
     +GXX32,GZE32,GZC32,GYLR33,GFXB33,GYY33,GXX33,GZE33,GZC33

      REAL GEAT(100,100),GOEF(1000,12)


      CHARACTER*1 SELECT,ANS
      CHARACTER DATAF*79,NAME*6

      INTEGER GPL,I,N,IB,GWIDE,GDEEP,COUNT,CONTEMP,ZER,GSEL,NMAX,TMAX,HT
     +RS,D1,D2,D3,D4,D5,D6,D7,MAXIT,LOCVAR,IC,ID,IE,IF

      INTEGER GIH(1000),GJH(1000),GCON(1000,12)
      CONTEMP=6
      ZER=0
```

```
      NMAX=750
      TMAX=50
      HTRS=6
      D1=2
      D2=4
      D3=6
      D4=0
      D5=0
      D6=0
      D7=0
      ACC=0.05
      DAMP=0.666667
      MAXIT=12
      CONFAC=0.8
      COUNT=6*GPL
      LOCVAR=COUNT


C
C
898   CALL CLS

      WRITE(*,803)
803   FORMAT(///,'     THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY
     + INTO THE',/,
     +'     EXISTING THERMAL ANALYZER, FURTHERMORE, THIS PROGRAM DOES',/,
     +'     NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE',/,
     +'     THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS    ',/,
     +'     PROGRAM.  THE FILE NAME IS LIMITED TO SIX CHARACTERS, AND',/,
     +'     SHOULD NOT HAVE ANY SPACES.                              ',///,
     +'     PLEASE ENTER THE DESIRED DATA FILE NAME:   ',2X )
      READ(*,804) NAME
804   FORMAT(A6)
552   WRITE(*,910)NAME
910   FORMAT(/////,'     YOU SELECTED ',A6,' FOR YOUR DATA FILE NAME
     +',//)
911   WRITE(*,811)
811   FORMAT('     IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N
     +FOR NO:    ',2X, )
      READ(*,812) ANS
812   FORMAT(A1)
C
      IF(ANS.EQ.'N') THEN
        GOTO 898
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'Y') THEN
        GOTO 897
      ELSE
         CALL CLS
         GOTO 552
897     ENDIF

C     ALLOW THE USER TO PROVIDE TITLE LINE FOR DATA FILE
```

```
833   CALL CLS
      WRITE(*,805)
805   FORMAT(////,'        ENTER THE DESIRED TITLE TO BE PLACED ON LINE
     +',/,'       NUMBER ONE OF THE OUTPUT DATA FILE:',///,
     +'          ',2X, )
      READ(*,806) DATAF
806   FORMAT(A79)
835   WRITE(*,831)
831   FORMAT(////,'        DO YOU WISH TO CHANGE THE TITLE OF YOUR OUTPU
     +T DATA FILE?',/,'        ENTER Y FOR YES AND N FOR NO:    ',2X, )
      READ(*,832) ANS
832   FORMAT(A1)
C
      IF(ANS.EQ.'Y') THEN
        GOTO 833
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'N') THEN
        GOTO 834
      ELSE
        GOTO 835
834     ENDIF
C

C*************************************************************************
C**************COEFFICIENTS FOR EPOXY AND COPPER LAYERS*****************
C*************************************************************************

C
      DO 90 I=1,GPL
      N=1
      IB=GPL+I
      IC=2*GPL+I
      ID=3*GPL+I
      IE=4*GPL+I
      IF=5*GPL+I

C
C*********************************CORNERS*********************************
C
      IF ((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I).EQ
     +.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP COPPER
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(ID,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
```

```
      GCON(IE,N) = 6
      GCON(IF,N) = 6
C
C
C    LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C    LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C    LEFT COEFFICIENT
      GOEF(I,N) = GYLR31
      GOEF(IC,N) = GYLR32
      GOEF(IE,N) = GYLR33
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      N=N+1
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
      GCON(IC,N) = 7551
      GCON(ID,N) = 7551
      GCON(IE,N) = 7551
      GCON(IF,N) = 7551

C
C    RIGHT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY

      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1

C
C    RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C    LEFT COEFFICIENT
C
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
```

```
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
C
C    RIGHT COEFFICIENT
C
      GOEF(I,N) = GYLR31
      GOEF(IC,N) = GYLR32
      GOEF(IE,N) = GYLR33
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      N=N+1
      GCON(I,N) = 7541
      GCON(IB,N) = 7541
      GCON(IC,N) = 7541
      GCON(ID,N) = 7541
      GCON(IE,N) = 7541
      GCON(IF,N) = 7541

      ENDIF
C
C    FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C    FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C    FRONT COEFFICIENT
      GOEF(I,N)=GFXB31
      GOEF(IC,N)=GFXB32
      GOEF(IE,N)=GFXB33
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(EF,N)=GFXB
      N=N+1
      GCON(I,N)  = 7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521
      GCON(IE,N) = 7521
      GCON(IF,N) = 7521
C    BACK COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
```

```
C     BACK EDGE
       ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
       GOEF(I,N)=GXX31
       GOEF(IC,N)=GXX32
       GOEF(IE,N)=GXX33
       GOEF(IB,N)=GXX
       GOEF(ID,N)=GXX
       GOEF(IF,N)=GXX
       N=N+1
       GCON(I,N)  = 10*(I-GWIDE)+1
       GCON(IB,N) = 10*(IB-GWIDE)+1
       GCON(IC,N) = 10*(IC-GWIDE)+1
       GCON(ID,N) = 10*(ID-GWIDE)+1
       GCON(IE,N) = 10*(IE-GWIDE)+1
       GCON(IF,N) = 10*(IF-GWIDE)+1

C     BACK COEFFICIENT
       GOEF(I,N)=GFXB31
       GOEF(IC,N)=GFXB32
       GOEF(IE,N)=GFXB33
       GOEF(IB,N)=GFXB
       GOEF(ID,N)=GFXB
       GOEF(IF,N)=GFXB
       N=N+1
       GCON(I,N) = 7511
       GCON(IB,N) = 7511
       GCON(IC,N) = 7511
       GCON(ID,N) = 7511
       GCON(IE,N) = 7511
       GCON(IF,N) = 7511

       ENDIF
C
C     TOP COEFFICIENT
       GOEF(I,N)=GZE31
       GOEF(IC,N)=GZE32
       GOEF(IE,N)=GZE33
       GOEF(IB,N)=GZC31
       GOEF(ID,N)=GZC32
       GOEF(IF,N)=GZC33
       N=N+1
       GCON(I,N) = 7511
       GCON(IC,N) = 7511
       GCON(IE,N) = 7511
       GCON(IB,N) = 10*I+1
       GCON(ID,N) = 10*IC+1
       GCON(IF,N) = 10*IE+1
C
C     BOTTOM COEFFICIENT
C
       GOEF(I,N) =  GZE31
       GOEF(IC,N) = GZE32
       GOEF(IE,N) = GZE33
       GOEF(IB,N) = GZC31
       GOEF(ID,N) = GZC32
```

```
      GOEF(IF,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1


C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF

C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE NUMBER OF CONNECTIONS FOR COPPER AND EPOXY LAYERS
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(ID,N) = 6
      GCON(IC,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
C
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
```

```
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
C
C     FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GFXB31
      GOEF(IC,N)=GFXB32
      GOEF(IE,N)=GFXB33
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521
      GCON(IE,N) = 7521
      GCON(IF,N) = 7521

C     BACK COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1

C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
```

```
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1

C    BACK COEFFICIENT
      GOEF(I,N)=GFXB31
      GOEF(IC,N)=GFXB32
      GOEF(IE,N)=GFXB33
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      GCON(IC,N) = 7511
      GCON(ID,N) = 7511
      GCON(IE,N) = 7511
      GCON(IF,N) = 7511
      ENDIF
C
C    TOP COEFFICIENT
      GOEF(I,N)=GZE31
      GOEF(IC,N)=GZE32
      GOEF(IE,N)=GZE33
      GOEF(IB,N)=GZC31
      GOEF(ID,N)=GZC32
      GOEF(IF,N)=GZC33

      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1


C
C    BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZE31
      GOEF(IC,N) = GZE32
      GOEF(IE,N) = GZE33
      GOEF(IB,N) = GZC31
      GOEF(ID,N) = GZC32
      GOEF(IF,N) = GZB

      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
```

```
      GCON(IF,N) = 10*(IF+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF

CXCXCXC
CXCXCXC
C
C
C     LEFT AND RIGHT EDGES EXCLUDING CORNERS
C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I)
     +.EQ.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
C
C
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
      GOEF(I,N) = GYLR31
      GOEF(IC,N) = GYLR32
      GOEF(IE,N) = GYLR33
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      N=N+1
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
      GCON(IC,N) = 7551
      GCON(ID,N) = 7551
      GCON(IE,N) = 7551
      GCON(IF,N) = 7551
C
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
```

```
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
C
C     RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
C
C     RIGHT COEFFICIENT
C
      GOEF(I,N)  = GYLR31
      GOEF(IC,N) = GYLR32
      GOEF(IE,N) = GYLR33
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      N=N+1
      GCON(I,N)  = 7541
      GCON(IB,N) = 7541
      GCON(IC,N) = 7541
      GCON(ID,N) = 7541
      GCON(IE,N) = 7541
      GCON(IF,N) = 7541
      ENDIF
C
C     FRONT COEFFICIENT
      GOEF(I,N) =GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
```

```
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
C    BACK COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
C
C    TOP COEFFICIENT
      GOEF(I,N)=GZE31
      GOEF(IC,N)=GZE32
      GOEF(ID,N)=GZE33
      GOEF(IB,N)=GZC31
      GOEF(ID,N)=GZC32
      GOEF(IF,N)=GZC33
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1
C
C    BOTTOM COEFFICIENT
C
      GOEF(I,N)  = GZE31
      GOEF(IC,N) = GZE32
      GOEF(IE,N) = GZE33

      GOEF(IB,N) = GZC31
      GOEF(ID,N) = GZC32
      GOEF(IF,N) = EZB
      N=N+1
      GCON(I,N)  = 10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1

C
C    HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
```

```
      GCON(I,N)=9991
      ENDIF
C
C
C
C****DETERMINE COEFFICIENTS FOR ALL NODES NOT TOUCHING AN EDGE*******
C
      ELSEIF((GIH(I).NE.1.OR.GIH(I).NE.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE CONNECTIONS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6



C
C
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I-1) + 1
      GCON(IB,N) = 10* (IB-1) + 1
      GCON(IC,N) = 10 * (IC-1) + 1
      GCON(ID,N) = 10 * (ID-1) + 1
      GCON(IE,N) = 10 * (IE-1) + 1
      GCON(IF,N) = 10 * (IF-1) + 1
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY31
      GOEF(IC,N) = GYY32
      GOEF(IE,N) = GYY33
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
```

```
      GCON(IF,N) = 10 * (IF+1) + 1
C
C    FRONT COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
C    BACK COEFFICIENT
      GOEF(I,N)=GXX31
      GOEF(IC,N)=GXX32
      GOEF(IE,N)=GXX33
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
C    TOP COEFFICIENT
      GOEF(I,N)=GZE31
      GOEF(IC,N)=GZE32
      GOEF(IE,N)=GZE33
      GOEF(IB,N)=GZC31
      GOEF(ID,N)=GZC32
      GOEF(IF,N)=GZC33
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(IC,N) = 7511
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1

C
C    BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZE31
      GOEF(IC,N) = GZE32
      GOEF(IE,N) = GZE33
      GOEF(IB,N) = GZC31
      GOEF(ID,N) = GZC32
      GOEF(IF,N) = GZC33
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
```

```
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
      ENDIF
90    CONTINUE
C
C     GENERATE DATA FILE
C
      OPEN (3,FILE=NAME,FORM='FORMATTED',ACCESS='DIRECT',RECL=108,STATUS
     +='NEW')
      WRITE(3,909) DATAF
909   FORMAT(1X,A79)
      WRITE(3,908) COUNT,CONTEMP,ZER,ZER,ZER,ZER,ZER,ZER,GSEL
908   FORMAT(2X,9(I3,5X))
      WRITE(3,907) ZER,ZER,ZER
907   FORMAT(2X,3(I3,5X))
      WRITE(3,9081) NMAX,TMAX,HTRS,D1,D2,D3,D4,D5,D6,D7
9081  FORMAT(2X,9(I3,5X))
      WRITE(3,905) ACC,DAMP,MAXIT,CONFAC,GIBT
905   FORMAT(1X,2(F9.7,1X),14X,I2,1X,F9.7,1X,F9.5)
      WRITE(3,906) GPRT,GFT,GBT,GRT,GLT,GWRT
906   FORMAT(1X,6(F12.3,1X))
      DO 112 I=1,LOCVAR
      WRITE(3,9100) GCON(I,1),GCON(I,2),GCON(I,3),GCON(I,4),GCON(I,5),GC
     +ON(I,6),GCON(I,7),GCON(I,8)
9100  FORMAT(I4,3X,7(I5,7X))
      WRITE(3,9110) GOEF(I,1),GOEF(I,2),GOEF(I,3),GOEF(I,4),GOEF(I,5),GO
     +EF(I,6),GOEF(I,7)
9110  FORMAT(7(F9.3,3X))
112   CONTINUE
      CLOSE (3)
      CALL CLS
      WRITE(*,999) NAME
999   FORMAT(////,'          THE OUTPUT DATA HAS BEEN PLACED IN A FILE
     +NAMED ',A6,/////,'                         <PRESS ENTER TO CONTINUE>')
      READ(*,5912) ANS
5912  FORMAT(A1)
      END
C**********************************************************************
C
C     TITLE:       MODEL BUILDER
C     SUBROUTINE:  S4
C     DATE:        09 JUL 91
C     AUTHOR:      LT STEVE GLASER
C     COMPILER:    MICROSOFT VERSION 4.01
C     LINKER:      MICROSOFT VERSION 3.55
C     SUBROUTINE S4 IS CALLED FROM SUBROUTINE PCBS3.  S4 GENERATES THE
```

```
C     OUTPUT DATA FILE FOR THE FOUR COPPER LAYER CASE.
C
C
       SUBROUTINE S4 (GYLR,GFXB,GYY,GXX,GZC41,GZC42,GZC43,GZC44,GZB.GYLR4
      +1,GYLR42,GYLR43,GYLR44,GFXB41,GFXB42,GFXB43,GFXB44,GYY41,GYY42,GYY
      +43,GYY44,GXX41,GXX42,GXX43,GXX44,GZE41,GZE42,GZE43,GZE44,GPL,GIH,G
      +JH,GEAT,GWIDE,GDEEP,GOEF,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT,GSEL)

       REAL GYLR,GFXB,GYY,GXX,GZC41,GZB,GYLR41,GFXB41,GYY41,GXX41,GZE41,A
      +CC,DAMP,CONFAC,GIBT,GPRT,GWRT,GRT,GLT,GFT,GBT,GYLR42,GFXB42,GYY42,
      +GXX42,GZE42,GZC42,GYLR43,GFXB43,GYY43,GXX43,GZE43,GZC43,GYLR44,GFX
      +B44,GYY44,GXX44,GZE44,GZC44

       REAL GEAT(100,100),GOEF(1000,12)


       CHARACTER*1 SELECT,ANS
       CHARACTER DATAF*79,NAME*6

       INTEGER GPL,I,N,IB,GWIDE,GDEEP,COUNT,CONTEMP,ZER,GSEL,NMAX,TMAX,HT
      +RS,D1,D2,D3,D4,D5,D6,D7,MAXIT,LOCVAR,IC,ID,IE,IF,IG,IH

       INTEGER GIH(1000),GJH(1000),GCON(1000,12)
       CONTEMP=6
       ZER=0
       NMAX=750
       TMAX=50
       HTRS=6
       D1=2
       D2=4
       D3=6
       D4=0
       D5=0
       D6=0
       D7=0
       ACC=0.05
       DAMP=0.666667
       MAXIT=12
       CONFAC=0.8
       COUNT=8*GPL
       LOCVAR=COUNT


C
C
898    CALL CLS

       WRITE(*,803)
803    FORMAT(///,'     THIS PROGRAM CREATES AN OUTPUT DATA FILE FOR ENTRY
      + INTO THE',/,
      +'     EXISTING THERMAL ANALYZER, FURTHERMORE, THIS PROGRAM DOES',/,
      +'     NOT ERASE OR WRITE OVER THE EXISTING DATA FILE. THEREFORE',/,
      +'     THE USER WILL NAME THE DATA FILE FOR EACH RUN OF THIS    ',/,
      +'     PROGRAM.  THE FILE NAME IS LIMITED TO SIX CHARACTERS, AND',/,
      +'     SHOULD NOT HAVE ANY SPACES.                              ',///,
      +'     PLEASE ENTER THE DESIRED DATA FILE NAME:  ',2X )
```

```
      READ(*,804) NAME
804   FORMAT(A6)
552   WRITE(*,910)NAME
910   FORMAT(/////,'    YOU SELECTED ',A6,' FOR YOUR DATA FILE NAME
     +',//)
911   WRITE(*,811)
811   FORMAT('    IS THIS THE DESIRED SELECTION?  ENTER Y FOR YES AND N
     +FOR NO:    ',2X, )
      READ(*,812) ANS
812   FORMAT(A1)
C
      IF(ANS.EQ.'N') THEN
        GOTO 898
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'Y') THEN
        GOTO 897
      ELSE
         CALL CLS
         GOTO 552
897     ENDIF

C     ALLOW THE USER TO PROVIDE TITLE LINE FOR DATA FILE

833   CALL CLS
      WRITE(*,805)
805   FORMAT(////,'          ENTER THE DESIRED TITLE TO BE PLACED ON LINE
     +',/,'        NUMBER ONE OF THE OUTPUT DATA FILE:',///,
     +'          ',2X, )
      READ(*,806) DATAF
806   FORMAT(A79)
835   WRITE(*,831)
831   FORMAT(////,'          DO YOU WISH TO CHANGE THE TITLE OF YOUR OUTPU
     +T DATA FILE?',/,'          ENTER Y FOR YES AND N FOR NO:    ',2X, )
      READ(*,832) ANS
832   FORMAT(A1)
C
      IF(ANS.EQ.'Y') THEN
        GOTO 833
      ELSE
        CONTINUE
      ENDIF
      IF(ANS.EQ.'N') THEN
        GOTO 834
      ELSE
        GOTO 835
834     ENDIF
C

C*************************************************************************
C**************COEFFICIENTS FOR EPOXY AND COPPER LAYERS*****************
C*************************************************************************

C
      DO 90 I=1,GPL
```

```
      N=1
      IB=GPL+I
      IC=2*GPL+I
      ID=3*GPL+I
      IE=4*GPL+I
      IF=5*GPL+I
      IG=6*GPL+I
      IH=7*GPL+I
C
C*********************************CORNERS********************************
C
      IF ((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I).EQ
     +.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP COPPER
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(ID,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
      GCON(IG,N) = 6
      GCON(IH,N) = 6


C
C
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
      GOEF(I,N)  = GYLR41
      GOEF(IC,N) = GYLR42
      GOEF(IE,N) = GYLR43
      GOEF(IG,N) = GYLR44
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      GOEF(IG,N) = GYLR
      GOEF(IH,N) = GYLR
      N=N+1
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
      GCON(IC,N) = 7551
      GCON(ID,N) = 7551
      GCON(IE,N) = 7551
      GCON(IF,N) = 7551
      GCON(IG,N) = 7551
```

```
      GCON(IH,N) = 7551

C
C    RIGHT COEFFICIENT
      GOEF(I,N) = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
      GCON(IG,N) = 10 * (IG+1) + 1
      GCON(IH,N) = 10 * (IH+1) + 1

C
C    RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C    LEFT COEFFICIENT
C
      GOEF(I,N) = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N)  = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
      GCON(IG,N) = 10*(IG-1)+1
      GCON(IH,N) = 10*(IH-1)+1
C
C    RIGHT COEFFICIENT
C
      GOEF(I,N)  = GYLR41
      GOEF(IC,N) = GYLR42
      GOEF(IE,N) = GYLR43
      GOEF(IG,N) = GYLR44
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      GOEF(IH,N) = GYLR
```

```
      N=N+1
      GCON(I,N) = 7541
      GCON(IB,N) = 7541
      GCON(IC,N) = 7541
      GCON(ID,N) = 7541
      GCON(IE,N) = 7541
      GCON(IF,N) = 7541
      GCON(IG,N) = 7541
      GCON(IH,N) = 7541
      ENDIF
C
C    FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C    FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C    FRONT COEFFICIENT
      GOEF(I,N)=GFXB41
      GOEF(IC,N)=GFXB42
      GOEF(IE,N)=GFXB43
      GOEF(IG,N)=GFXB44
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      GOEF(IH,N)=GFXB
      N=N+1
      GCON(I,N)  = 7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521
      GCON(IE,N) = 7521
      GCON(IF,N) = 7521
      GCON(IG,N) = 7521
      GCON(IH,N) = 7521
C    BACK COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1


C    BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C    FRONT COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
```

```
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N)  = 10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
      GCON(IG,N) = 10*(IG-GWIDE)+1
      GCON(IH,N) = 10*(IH-GWIDE)+1
C     BACK COEFFICIENT
      GOEF(I,N)=GFXB41
      GOEF(IC,N)=GFXB42
      GOEF(IE,N)=GFXB43
      GOEF(IG,N)=GFXB44
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      GOEF(IH,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      GCON(IC,N) = 7511
      GCON(ID,N) = 7511
      GCON(IE,N) = 7511
      GCON(IF,N) = 7511
      GCON(IG,N) = 7511
      GCON(IH,N) = 7511
      ENDIF
C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE41
      GOEF(IC,N)=GZE42
      GOEF(IE,N)=GZE43
      GOEF(IG,N)=GZE44
      GOEF(IB,N)=GZC41
      GOEF(ID,N)=GZC42
      GOEF(IF,N)=GZC43
      GOEF(IH,N)=GZC44
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IE,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
      GCON(IF,N) = 10*IE+1
      GCON(IG,N) = 7511
      GCON(IH,N) = 10*IF+1
C
C     BOTTOM COEFFICIENT
C
```

```
      GOEF(I,N) =  GZE41
      GOEF(IC,N) = GZE42
      GOEF(IE,N) = GZE43
      GOEF(IG,N) = GZE44
      GOEF(IB,N) = GZC41
      GOEF(ID,N) = GZC42
      GOEF(IF,N) = GZC43
      GOEF(IH,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1
      GCON(IG,N) = 10*(IG+GPL)+1
      GCON(IH,N) = 10*(IH+GPL)+1
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF

C
      ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE NUMBER OF CONNECTIONS FOR COPPER AND EPOXY LAYERS
      IF (GEAT(GIH(I),GJH(I)).EQ.0.0) THEN
          GCON(I,N) = 6
      ELSE
          GCON(I,N) = 7
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(ID,N) = 6
      GCON(IC,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
      GCON(IG,N) = 6
      GCON(IH,N) = 6
C
C     LEFT AND RIGHT COEFFICIENTS
C     LEFT COEFFICIENT
      GOEF(I,N)  = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
```

```
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
      GCON(IG,N) = 10*(IG-1)+1
      GCON(IH,N) = 10*(IH-1)+1
C
C    RIGHT COEFFICIENT
      GOEF(I,N)  = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
      GCON(IG,N) = 10 * (IG+1) + 1
      GCON(IH,N) = 10 * (IH+1) + 1
C
C    FRONT AND BACK COEFFICIENTS DEPENDING ON WHICH EDGE
C
C    FRONT EDGE
      IF (GIH(I).EQ.1) THEN
C    FRONT COEFFICIENT
      GOEF(I,N)=GFXB41
      GOEF(IC,N)=GFXB42
      GOEF(IE,N)=GFXB43
      GOEF(IG,N)=GFXB44
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      GOEF(IH,N)=GFXB
      N=N+1
      GCON(I,N) =7521
      GCON(IB,N) = 7521
      GCON(IC,N) = 7521
      GCON(ID,N) = 7521
      GCON(IE,N) = 7521
      GCON(IF,N) = 7521
      GCON(IG,N) = 7521
      GCON(IH,N) = 7521

C    BACK COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
```

```
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
      GCON(IG,N) = 10*(IG+GWIDE)+1
      GCON(IH,N) = 10*(IH+GWIDE)+1
C     BACK EDGE
      ELSEIF (GIH(I).EQ.GDEEP) THEN
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
      GCON(IG,N) = 10*(IE-GWIDE)+1
      GCON(IH,N) = 10*(IF-GWIDE)+1
C     BACK COEFFICIENT
      GOEF(I,N)=GFXB41
      GOEF(IC,N)=GFXB42
      GOEF(IE,N)=GFXB43
      GOEF(IG,N)=GFXB44
      GOEF(IB,N)=GFXB
      GOEF(ID,N)=GFXB
      GOEF(IF,N)=GFXB
      GOEF(IH,N)=GFXB
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 7511
      GCON(IC,N) = 7511
      GCON(ID,N) = 7511
      GCON(IE,N) = 7511
      GCON(IF,N) = 7511
      GCON(IG,N) = 7511
      GCON(IH,N) = 7511
      ENDIF
C
C     TOP COEFFICIENT
      GOEF(I,N)=GZE41
      GOEF(IC,N)=GZE42
      GOEF(IE,N)=GZE43
```

```
      GOEF(IG,N)=GZE44
      GOEF(IB,N)=GZC41
      GOEF(ID,N)=GZC42
      GOEF(IF,N)=GZC43
      GOEF(IH,N)=GZC44
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1
      GCON(IG,N) = 7511
      GCON(IH,N) = 10*IG+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZE41
      GOEF(IC,N) = GZE42
      GOEF(IE,N) = GZE43
      GOEF(IG,N) = GZE44
      GOEF(IB,N) = GZC41
      GOEF(ID,N) = GZC42
      GOEF(IF,N) = GZC43
      GOEF(IH,N) = GZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1
      GCON(IG,N) = 10*(IG+GPL)+1
      GCON(IH,N) = 10*(IH+GPL)+1
C
C     HEAT INPUT
       IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
       GOEF(I,N) = GEAT(GIH(I),GJH(I))
       N=N+1
       GCON(I,N)=9991
       ENDIF

CXCXCXC
CXCXCXC
C
C
C     LEFT AND RIGHT EDGES EXCLUDING CORNERS
C
       ELSEIF((GIH(I).EQ.1.OR.GIH(I).EQ.GDEEP).AND.(GJH(I).EQ.1.OR.GJH(I)
     +.EQ.GWIDE)) THEN
C
C     DETERMINE COEFFICIENTS FOR TOP LAYER
       IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
           GCON(I,N) = 7
       ELSE
           GCON(I,N) = 6
```

```
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
      GCON(IG,N) = 6
      GCON(IH,N) = 6
C
C     LEFT AND RIGHT COEFFICIENTS DEPENDING ON WHICH EDGE
C
C     LEFT EDGE
      IF (GJH(I).EQ.1) THEN
C     LEFT COEFFICIENT
      GOEF(I,N) = GYLR41
      GOEF(IC,N) = GYLR42
      GOEF(IE,N) = GYLR43
      GOEF(IG,N) = GYLR44
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      GOEF(IH,N) = GYLR
      N=N+1
      GCON(I,N) = 7551
      GCON(IB,N) = 7551
      GCON(IC,N) = 7551
      GCON(ID,N) = 7551
      GCON(IE,N) = 7551
      GCON(IF,N) = 7551
      GCON(IG,N) = 7551
      GCON(IH,N) = 7551
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
      GCON(IG,N) = 10 * (IG+1) + 1
      GCON(IH,N) = 10 * (IH+1) + 1
C     RIGHT EDGE
      ELSEIF (GJH(I).EQ.GWIDE) THEN
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY41
```

```
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10*(I-1)+1
      GCON(IB,N) = 10*(IB-1)+1
      GCON(IC,N) = 10*(IC-1)+1
      GCON(ID,N) = 10*(ID-1)+1
      GCON(IE,N) = 10*(IE-1)+1
      GCON(IF,N) = 10*(IF-1)+1
      GCON(IG,N) = 10*(IG-1)+1
      GCON(IH,N) = 10*(IH-1)+1
C     RIGHT COEFFICIENT
C
      GOEF(I,N)  = GYLR41
      GOEF(IC,N) = GYLR42
      GOEF(IE,N) = GYLR43
      GOEF(IG,N) = GYLR44
      GOEF(IB,N) = GYLR
      GOEF(ID,N) = GYLR
      GOEF(IF,N) = GYLR
      GOEF(IH,N) = GYLR
      N=N+1
      GCON(I,N)  = 7541
      GCON(IB,N) = 7541
      GCON(IC,N) = 7541
      GCON(ID,N) = 7541
      GCON(IE,N) = 7541
      GCON(IF,N) = 7541
      GCON(IG,N) = 7541
      GCON(IH,N) = 7541
      ENDIF
C
C     FRONT COEFFICIENT
      GOEF(I,N) =GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
      GCON(IG,N) = 10*(IG-GWIDE)+1
      GCON(IH,N) = 10*(IH-GWIDE)+1
C     BACK COEFFICIENT
```

```
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
      GCON(IG,N) = 10*(IG+GWIDE)+1
      GCON(IH,N) = 10*(IH+GWIDE)+1
C     TOP COEFFICIENT
      GOEF(I,N)=GZE41
      GOEF(IC,N)=GZE42
      GOEF(IE,N)=GZE43
      GOEF(IG,N)=GZE44
      GOEF(IB,N)=GZC41
      GOEF(ID,N)=GZC42
      GOEF(IF,N)=GZC43
      GOEF(IH,N)=GZC44
      N=N+1
      GCON(I,N) = 7511
      GCON(IC,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1
      GCON(IG,N) = 7511
      GCON(IH,N) = 10*IG+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N)  = GZE41
      GOEF(IC,N) = GZE42
      GOEF(IE,N) = GZE43
      GOEF(IG,N) = GZE44
      GOEF(IB,N) = GZC43
      GOEF(ID,N) = GZC43
      GOEF(IF,N) = GZC43
      GOEF(IH,N) = EZB
      N=N+1
      GCON(I,N)  = 10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1
      GCON(IG,N) = 10*(IG+GPL)+1
```

```
      GCON(IH,N) = 10*(IH+GPL)+1
C
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
C
C****DETERMINE COEFFICIENTS FOR ALL NODES NOT TOUCHING AN EDGE*******
C
      ELSEIF((GIH(I).NE.1.OR.GIH(I).NE.GDEEP).AND.(GJH(I).NE.1.OR.GJH(I)
     +.NE.GWIDE)) THEN
C
C     DETERMINE CONNECTIONS FOR TOP LAYER
      IF (GEAT(GIH(I),GJH(I)).NE.0.0) THEN
          GCON(I,N) = 7
      ELSE
          GCON(I,N) = 6
      ENDIF
C
C     CONNECTIONS FOR EPOXY LAYER
      GCON(IB,N) = 6
      GCON(IC,N) = 6
      GCON(ID,N) = 6
      GCON(IE,N) = 6
      GCON(IF,N) = 6
      GCON(IG,N) = 6
      GCON(IH,N) = 6
C
C
C     LEFT COEFFICIENT
      GOEF(I,N) = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N)  = 10 * (I-1) + 1
      GCON(IB,N) = 10* (IB-1) + 1
      GCON(IC,N) = 10 * (IC-1) + 1
      GCON(ID,N) = 10 * (ID-1) + 1
      GCON(IE,N) = 10 * (IE-1) + 1
      GCON(IF,N) = 10 * (IF-1) + 1
      GCON(IG,N) = 10 * (IG-1) + 1
      GCON(IH,N) = 10 * (IH-1) + 1
C     RIGHT COEFFICIENT
      GOEF(I,N) = GYY41
      GOEF(IC,N) = GYY42
      GOEF(IE,N) = GYY43
      GOEF(IG,N) = GYY44
      GOEF(IB,N) = GYY
      GOEF(ID,N) = GYY
```

```
      GOEF(IF,N) = GYY
      GOEF(IH,N) = GYY
      N=N+1
      GCON(I,N) = 10 * (I+1) + 1
      GCON(IB,N) = 10* (IB+1) + 1
      GCON(IC,N) = 10 * (IC+1) + 1
      GCON(ID,N) = 10 * (ID+1) + 1
      GCON(IE,N) = 10 * (IE+1) + 1
      GCON(IF,N) = 10 * (IF+1) + 1
      GCON(IG,N) = 10 * (IG+1) + 1
      GCON(IH,N) = 10 * (IH+1) + 1
C
C     FRONT COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IG,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I-GWIDE)+1
      GCON(IB,N) = 10*(IB-GWIDE)+1
      GCON(IC,N) = 10*(IC-GWIDE)+1
      GCON(ID,N) = 10*(ID-GWIDE)+1
      GCON(IE,N) = 10*(IE-GWIDE)+1
      GCON(IF,N) = 10*(IF-GWIDE)+1
      GCON(IG,N) = 10*(IG-GWIDE)+1
      GCON(IH,N) = 10*(IH-GWIDE)+1
C     BACK COEFFICIENT
      GOEF(I,N)=GXX41
      GOEF(IC,N)=GXX42
      GOEF(IE,N)=GXX43
      GOEF(IH,N)=GXX44
      GOEF(IB,N)=GXX
      GOEF(ID,N)=GXX
      GOEF(IF,N)=GXX
      GOEF(IH,N)=GXX
      N=N+1
      GCON(I,N) =10*(I+GWIDE)+1
      GCON(IB,N) = 10*(IB+GWIDE)+1
      GCON(IC,N) = 10*(IC+GWIDE)+1
      GCON(ID,N) = 10*(ID+GWIDE)+1
      GCON(IE,N) = 10*(IE+GWIDE)+1
      GCON(IF,N) = 10*(IF+GWIDE)+1
      GCON(IG,N) = 10*(IG+GWIDE)+1
      GCON(IH,N) = 10*(IH+GWIDE)+1
C     TOP COEFFICIENT
      GOEF(I,N)=GZE41
      GOEF(IC,N)=GZE42
      GOEF(IE,N)=GZE43
      GOEF(IG,N)=GZE44
      GOEF(IB,N)=GZC41
      GOEF(ID,N)=GZC42
      GOEF(IF,N)=GZC43
```

```
      GOEF(IH,N)=GZC44
      N=N+1
      GCON(I,N) = 7511
      GCON(IB,N) = 10*I+1
      GCON(IC,N) = 7511
      GCON(ID,N) = 10*IC+1
      GCON(IE,N) = 7511
      GCON(IF,N) = 10*IE+1
      GCON(IG,N) = 7511
      GCON(IH,N) = 10*IG+1
C
C     BOTTOM COEFFICIENT
C
      GOEF(I,N) =  GZE41
      GOEF(IC,N) = GZE42
      GOEF(IE,N) = GZE43
      GOEF(IG,N) = GZE44
      GOEF(IB,N) = GZC41
      GOEF(ID,N) = GZC42
      GOEF(IF,N) = GZC43
      GOEF(IH,N) = EZB
      N=N+1
      GCON(I,N) =10*(I+GPL)+1
      GCON(IB,N) = 10*(IB+GPL)+1
      GCON(IC,N) = 10*(IC+GPL)+1
      GCON(ID,N) = 10*(ID+GPL)+1
      GCON(IE,N) = 10*(IE+GPL)+1
      GCON(IF,N) = 10*(IF+GPL)+1
      GCON(IG,N) = 10*(IG+GPL)+1
      GCON(IH,N) = 10*(IH+GPL)+1
C     HEAT INPUT
      IF(GEAT(GIH(I),GJH(I)).NE.0.0) THEN
      GOEF(I,N) = GEAT(GIH(I),GJH(I))
      N=N+1
      GCON(I,N)=9991
      ENDIF
      ENDIF
90    CONTINUE
C
C     GENERATE DATA FILE
C
      OPEN (3,FILE=NAME,FORM='FORMATTED',ACCESS='DIRECT',RECL=108,STATUS
     +='NEW')
      WRITE(3,909) DATAF
909   FORMAT(1X,A79)
      WRITE(3,908) COUNT,CONTEMP,ZER,ZER,ZER,ZER,ZER,ZER,GSEL
908   FORMAT(2X,9(I3,5X))
      WRITE(3,907) ZER,ZER,ZER
907   FORMAT(2X,3(I3,5X))
      WRITE(3,9081) NMAX,TMAX,HTRS,D1,D2,D3,D4,D5,D6,D7
9081  FORMAT(2X,9(I3,5X))
      WRITE(3,905) ACC,DAMP,MAXIT,CONFAC,GIBT
905   FORMAT(1X,2(F9.7,1X),14X,I2,1X,F9.7,1X,F9.5)
      WRITE(3,906) GPRT,GFT,GBT,GRT,GLT,GWRT
906   FORMAT(1X,6(F12.3,1X))
      DO 112 I=1,LOCVAR
```

```
      WRITE(3,9100) GCON(I,1),GCON(I,2),GCON(I,3),GCON(I,4),GCON(I,5),GC
     +ON(I,6),GCON(I,7),GCON(I,8)
9100  FORMAT(I4,3X,7(I5,7X))
      WRITE(3,9110) GOEF(I,1),GOEF(I,2),GOEF(I,3),GOEF(I,4),GOEF(I,5),GO
     +EF(I,6),GOEF(I,7)
9110  FORMAT(7(F9.3,3X))
112   CONTINUE
      CLOSE (3)
      CALL CLS
      WRITE(*,999) NAME
999   FORMAT(////,'                        THE OUTPUT DATA HAS BEEN PLACED IN A FILE
     +NAMED ',A6,/////,'                              <PRESS ENTER TO CONTINUE>')
      READ(*,5912) ANS
5912  FORMAT(A1)
      END
C ***************************************************************************
```

LIST OF REFERENCES

1. Kraus, A.D. and Bar-Cohen, A. *Thermal Analysis and Control of Electronic Equipment* , pp 1-97,555-568, Hemisphere Publishing Corporation, 1983.

2. Tummala, R.R. and Rymaszewski, E.J. *Microelectronics Packaging Handbook*, pp 853-917, Van Nostrand Reinhold Limited, 1989.

3. Frankel, E.G. *Systems Reliability and Risk Analysis* , pp 16-20, Kluwer Academic Publishers, 1988.

4. Keonjian, E. *Microelectronics*, pp 71-75, McGraw Hill Book Company Inc., 1963.

5. Tipler, P.A. *Physics* , pp 530-538, Worth Publishers, 1982.

6. Roesch, P.K. *Development of a Model Builder for a Microcircuit Substrate*,, Naval Postgraduate School, Monterey CA, June 1991.

7. Obert, E. F. , and Young, R. L. *Elements of Thermodynamics and Heat Transfer*, pp 370-434, McGraw-Hill Book Company, Inc., 1962.

8. Willheim, J. A., *Computer Aided Thermal Analysis of Microcircuit Structure*, Naval Postgraduate School, Monterey CA, December 1990.

9. Kraus, A.D. *Thermal Analysis of Electronic Equipment*,, Intercept Software, Campbell, CA, June 1985.